UNIVERSITÀ DEGLI STUDI DI PISA

Facoltà di Ingegneria

Corso di Laurea Specialistica in Ingegneria Informatica

Progetto ed implementazione di un modulo software per la realizzazione di connessioni MPLS inter-dominio con garanzie di qualità del servizio

Il candidato:

Paolo Sozzi

I relatori:

Luciano Lenzini

Enzo Mingozzi

Anno Accademico 2005/2006

URN	etd-02132007-183558
Autore	Sozzi, Paolo
Indirizzo email	pavelo.s@tin.it
Struttura	INGEGNERIA, FACOLTA'
Corso di studi	INGEGNERIA INFORMATICA
Relatore	- Luciano Lenzini - Enzo Mingozzi
Tipo di tesi	LS
Titolo	Progetto ed implementazione di un modulo software per la realizzazione di connessioni MPLS inter-dominio con garanzie di qualità del servizio
Data inizio appello	2007-03-05
Data presentazione	2007-02-13

A mia madre

Che mi ha permesso di arrivare fino a qui.

Paolo

Ringraziamenti

Per prima cosa vorrei ringraziare tutti coloro che in un modo o in un altro hanno contribuito a questo lavoro:

- La mia famiglia, per tutto il supporto che mi hanno dato in questi anni e in special modo durante il periodo di tesi.
- Il Prof. Luciano Lenzini, per la possibilità che mi ha dato di lavorare a questo progetto.
- Il Prof. Enzo Mingozzi, per la disponibilità con cui mi ha seguito durante tutto il periodo della tesi.
- Tutti i membri del gruppo che ha lavorato allo sviluppo di TERO, in particolar modo Giovanni Stea per il lavoro di progettazione, Luca Bisti per il lavoro di implementazione e Simone Bisogni per avermi fatto da Cicerone.
- La mia ragazza per la pazienza che ha dimostrato nei miei confronti in questi mesi.
- I compagni di casa, in particolare Fabio, Dario e Michele, con cui ho condiviso l'esperienza universitaria, per avermi sopportato in questi anni.
- Tutti i ragazzi del laboratorio rosso e del PerLab, con cui ho condiviso gli sfoghi, le ansie, ma anche le gioie e le piccole soddisfazioni del periodo di tesi.

- I ragazzi della facoltà, con cui ha condiviso tutta, o parte dell'avventura universitaria, dalle lezioni agli esami, dalla mensa ai pranzi in Piazza dei Miracoli.
- I compagni del liceo che hanno scelto l'Università di Pisa, con i quali ho condiviso il tempo trascorso lontano dai libri in questi anni.
- I ragazzi di Follonica (e dintorni) che frequento.
- Gli amici che Travale che conosco da quando ero piccolo.
- Gli amici di Sempre: Chiara, Francesca, Lorenzo e Michele, con cui ho condiviso una vita intera.
- Tutti quelli che mi sono dimenticato di nominare, ma che meritano comunque un grazie.

Grazie di cuore a tutti.

Sommario

Indice delle figure

Tabella degli acronimi

AS	Autonomous System
BGP	Border Gateway Protocol
CAC	Call Admission Control
DSCP	Differentiated Services Code Point
ERO	Explicit Route Object
EuQoS	End-to-end QoS support over heterogeneous networks
FEC	Forwarding Equivalent Class
GMPLS	Generic Multi Protocol Label Switching
HGSO	Hard Guarantees Solution Option
IPTD	IP Transit Delay
IPDV	IP Delay Variation
IPLR	IP Loss Rate
IS-IS	Intermediate System - Intermediate System
ISP	Internet Service Provider
LGSO	Loose Guarantees Solution Option
l-QC	Local QC
LSP	Label Switched Path
LSR	Label Switched Router
Mescal	Management of End-to-end Quality of Service Across the Internet at Large
MPLS	Multi Protocol Label Switching
MPLS-TE	MPLS Traffic Engineering
OSPF	Open Shortest Path First
PHB	Per Hop Behavior
PCC	Path Computation Client
PCE	Path Computation Element

PCID	Path Computation ID
PCP	Path Communication Protocol
PRID	Path Reference ID
PSTN	Public Switched Telecommunication Network
QC	QoS Class
QoS	Quality of Service
QoS_NLRI	QoS related Network Reachability Information
RA	Resource Allocator
RIB	Routing Information Base
RM	Resource Manager
RM-DB	Resource Manager Data Base
RSVP	Resource Reservation Protocol
SGSO	Statistical Guarantees Solution Option
SIP	Session Initialization Protocol
SLA	Service Level Agreement
SLS	Service Level Specification
SLS-T	Service Level Specification Template
TE	Traffic Engineering
TED	Traffic Engineering Database
TERO	Traffic Engineering and Resource Optimization
VoIP	Voice over IP

1 Introduzione

Gli ultimi anni sono stati testimoni della continua crescita della richiesta di servizi che presuppongono la connettività alla rete Internet. Questa crescita ha portato con se un notevole aumento dello sviluppo di applicazioni multimediali, e sistemi di gestione dei dati. Queste nuove tecnologie, sempre più frequentemente, fanno riferimento al concetto di Qualità del Servizio; il modello di Internet così come lo conosciamo adesso però non offre supporto a tali richieste.

È necessario quindi, cambiare il concetto di Internet basato sul principio del Best Effort per garantire la Qualità del Servizio (***QoS – Quality of Service***). A tale scopo gruppi di ricerca, aziende ed enti del settore stanno sviluppando architetture, che offriranno un supporto nativo alla QoS. Tali architetture saranno la base, o il fondamento, di quella che attualmente è conosciuta come la prossima generazione di Internet. La QoS infatti sarà vista come parte integrante della rete stessa e non più come strumento opzionale.

La rete Internet è costituita da vari operatori, ognuno dei quali amministra la propria rete, secondo le proprie necessità e seguendo i propri criteri, in maniera completamente autonoma dagli altri. L'insieme delle reti sotto il controllo di un unico amministratore viene identificato come un dominio amministrativo. La maggior parte della ricerca basata sulla qualità del servizio si concentra sull'offerta di QoS limitata però ad un solo dominio. La possibilità di creare un supporto di QoS end-to-end alle applicazioni implica infatti che domini gestiti da organizzazioni differenti cooperino al fine di ottenere il livello di servizio richiesto.

Attualmente esistono strumenti che permettono di offrire QoS ma con il vincolo di non superare i confini del dominio amministrativo. Al di fuori di un dominio infatti, non è possibile conoscere a priori il trattamento che riceverà un determinato flusso di traffico e quindi diventa impossibile offrire QoS.

Il problema che ci troviamo a risolvere è come offrire QoS attraverso più domini amministrativi in modo tale da tenere in considerazione la struttura disomogenea di Internet. Si deve quindi creare un'architettura che permetta la cooperazione tra domini/***AS – Autonomous System*** differenti, così da fare in modo che un AS possa informare il suo vicino riguardo alle garanzie di QoS richieste per il traffico associato ad un determinato servizio. Allo stesso momento dobbiamo far si che l'altro AS possa controllare se il nuovo flusso non vada a compromettere le garanzie offerte precedentemente agli altri e, di conseguenza, decidere se accettare o meno il nuovo flusso.

Per permettere la cooperazione tra più domini amministrativi c'è la necessita di avere una base in comune, ovvero un accordo stipulato tra AS adiacenti in cui si specifichi il livello del servizio che un AS (nel ruolo di provider) offre ad un altro (nel ruolo di customer). Tale accordo è chiamato ***Service Level Agreement*** (***SLA***), e a sua volta ha una corrispondenza con un accordo di più basso livello detto ***Service Level Specification*** (***SLS***), in cui è specificato il profilo del traffico che l'AS provider può accettare dall'AS customer, e le garanzie del servizio che vengono offerte a tale traffico.

Un'architettura che affronta il problema sopra citato è stata sviluppata nel progetto europeo ***EuQoS - end-to-end QoS support over heterogeneous networks***. Tale progetto, iniziato nel settembre 2004, ha come obiettivo ricercare, integrare, testare, validare e dimostrare tecnologie di QoS per offrire infrastrutture di supporto ad applicazioni "QoS-Aware" complesse su domini vari ed eterogenei appartenenti ad enti scientifici, industriali e di ricerca.

Il lavoro svolto durante la tesi si inserisce all'interno del progetto EuQoS; più precisamente fa parte di un modulo interno del progetto chiamato ***TERO - Traffic***

Engineering and Resource Optimization, il cui compito è gestire l'allocazione delle risorse all'interno dell'architettura di EuQoS. L'obiettivo della tesi è la progettazione e l'implementazione di un protocollo di comunicazione che permetta di creare degli cammini con granularità di AS al fine di creare dei link virtuali attraverso i vari domini, che permettano la creazione di una overlay network capace di ridurre le comunicazioni dovute alla segnalazione nella rete, e semplificare la gestione delle risorse tramite una struttura gerarchica.

La procedura utilizzata per la creazione dei link sopra citati viene effettuata in più fasi, aumentando il livello di dettaglio. In primo luogo, come specificato precedentemente, si cerca un cammino tra i vari AS detto AS path. Tale cammino viene ricercato sulla rete essendo a conoscenza di:

- AS di partenza.
- AS di destinazione.
- Classe di servizio per il link in fase di costruzione.

Successivamente alla ricerca dell'AS path, sulla base di quest'ultimo si cerca un cammino più preciso, specificando tutti i router che sono attraversati, infine si prova ad installare il cammino sui router selezionati tramite il protocollo RSVP-TE.

Il cammino creato è un ***Label Switched Path*** (***LSP***), ovvero un cammino di tipo end-to-end in cui le operazioni di instradamento sono fatte attraverso il protocollo ***MPLS – Multi Protocol Label Switching***.

Dal punto di vista del cliente questa infrastruttura permette di ampliare il raggio di applicazione di tutti quei servizi che richiedono determinate garanzie per un corretto funzionamento. Dal punto di vista di chi offre il servizio, la possibilità di raggiungere più clienti può essere tradotta in un aumento degli stessi e di conseguenza dei possibili guadagni.

2 Stato dell'arte

Negli ultimi dieci anni si è potuto osservare una notevole integrazione tra computer e telefonia, sia sul lato degli apparecchi, sia sul lato delle infrastrutture. Sempre più frequentemente si preferisce utilizzare una rete a pacchetto rispetto ad una tradizionale rete di telecomunicazione PSTN[1]. L'arrivo della telefonia sulla rete Internet ha portato con se un forte aumento della presenza di programmi con requisiti di qualità del servizio. Basti pensare a telefonate VoIP, video conferenze, servizi di messaging, gaming on-line, oltre alle applicazioni già esistenti. Tali applicazioni per funzionare correttamente, o almeno sufficientemente bene da considerarsi utilizzabili hanno bisogno di avere delle garanzie del servizio da parte della rete sottostante.

Nonostante ciò l'attuale Internet continua ad offrire un servizio di tipo Best-Effort. Questo significa che la qualità che la rete offre non è predicibile. Essa varia a seconda, sia della posizione geografica dei due interlocutori, sia dall'istante temporale in cui avviene la comunicazione. Il paradigma Best-Effort non è in grado quindi di fornire un servizio ad un utente (o ad un'applicazione) che chiede una seppur minima garanzia.

Alcune applicazioni possono tuttavia continuare a funzionare anche senza un supporto alla QoS, seppur con possibili degradi delle prestazioni. Altre applicazioni invece, necessitano di garanzie stringenti al fine di un corretto funzionamento. In questi casi nasce il bisogno di tecnologie e strumenti creati

[1] Public Switched Telecommunication Network

appositamente per supportare la qualità del servizio. Strumenti che forniscano al paradigma Best-Effort la capacità di poter garantire la QoS da loro richiesta.

Risolvere questo problema resta però un compito assai complesso, specialmente se si considerano fattori fino ad adesso trascurati quali la disomogeneità della rete Internet, la semplicità dello sviluppo di protocolli e applicazioni, la disponibilità di risorse e mezzi, senza escludere il fattore economico.

2.1 Il progetto Mescal

Mescal – Management of End-to-end Quality of Service Across the Internet at Large è un progetto fondato parzialmente dall'Unione Europea il cui obiettivo consiste in: "*to propose and validate scalable, incremental solutions, enabling flexible deployment and delivery of inter-domain QoS across the Internet at large*".

Nel modello di Mescal un customer sottoscrive un accordo basato su QoS con il suo provider. Tale accordo è delineato sulla base di un SLA. L'obiettivo di Mescal sono le relazioni che si instaurano tra customer e provider, oppure tra provider differenti. Queste relazioni sono delineate sulla base di accordi detti SLA, e più specificatamente sull'aspetto tecnico degli SLA, ovvero gli SLS.

2.1.1 Servizi di connettività

Mescal è incentrato su servizi di connettività basati su QoS. Un servizio di connettività è un servizio che "passa attraverso" la rete per raggiungere particolari destinazioni partendo dai punti specificati, nel campo degli indirizzi IP. Gli aspetti di QoS dei servizi di connettività sono principalmente legati alla qualità tramite cui i datagrammi IP trasmessi dall'utente sono trasferiti sulla rete tra le due parti della comunicazione. I livelli più alti, specifici dei servizi offerti dalle applicazioni, e.g. i servizi di streaminig o video-on-demand sono al di fuori del obiettivo di Mescal.

Da notare il fatto che questi ultimi servizi spesso richiedono una dimensione della connessione, la quale, se non è fornita opportunamente, fa si che l'intero servizio non venga fornito affatto. Quindi i servizi di connettività devono essere studiati a priori, prima di installarci sopra altri servizi di più alto livello.

In Mescal i servizi di connettività sono distinti tra elementari e complessi. I servizi elementari sono strettamente di tipo punto-punto e unidirezionali, mentre i servizi complessi possono essere multi punto-multi punto e bidirezionali. Quindi i servizi complessi inglobano più servizi elementari a seconda di ciò che chiede il contesto a cui sono applicati. Allo stesso modo i servizi elementari possono essere visti come la base, o il nucleo, di un servizio di tipo complesso. Esempi tipici di servizi complessi sono i servizi di connettività attualmente offerti ai customer. Esempi concreti possono essere le VPN, l'accesso a internet, servizi di accesso a server. I servizi semplici invece possono solo esistere nel contesto dei servizi complessi, e tali non possono essere offerti al customer. Quindi il termine di servizio di connettività utilizzato implica un servizio di tipo complesso.

2.1.2 La qualità del servizio

Il termine servizio denota, dal punto di vista del customer, una specifica offerta fatta da un provider, la quale descrive in maniera chiara e non ambigua cosa offre, i termini e le condizioni sotto le quali può essere utilizzato. Allo stesso modo dal punto di vista del provider, un servizio denota un sottoinsieme delle potenzialità del dominio provider, con una chiara descrizione riguardante cosa e come tali risorse possano essere utilizzate dal customer o da terze parti in generale.

Il termine "servizio basato su QoS" o, più semplicemente, "servizio di QoS", indica un servizio che implica un valore aggiunto al customer.

2.1.2.1 I Service Level Agreement

I sevizi che sono attualmente offerti ai customer, sono offerti sulla base accordi tra le parti, noti con il termine di ***Service Level Agreement*** (***SLA***). Gli SLA sono instaurati tra customer e provider e descrivono le caratteristiche del servizio e le

responsabilità che l'uno ha verso l'altro e viceversa per l'utilizzo o l'offerta del servizio stesso. L'approvazione di SLA da parte di customer e provider prevede che, da un lato, gli SLA siano descritti in maniera sufficientemente comprensibile, al fine di essere compresi dal customer, dall'altro, che siano assicurate le caratteristiche dei servizi così come espresso negli SLA; siano cioè realmente offerte così come sono state accettate.

Gli SLA possono essere stabiliti anche tra due provider, con un provider che svolge il ruolo di customer verso il secondo, e l'altro che svolge il ruolo di customer nella direzione del primo; al fine di aumentare reciprocamente la raggiungibilità dei loro rispettivi servizi. SLA tra provider estendono il concetto di "*peering business agreement*" che esiste oggi tra i provider per scambiarsi reciprocamente traffico alla data frequenza. Ovviamente in un Internet che non offre QoS questi accordi non hanno utilità, laddove invece si vuole offrire QoS, c'è il bisogno di includere le descrizioni delle caratteristiche del servizio, tenendo in considerazione quest'ultime, ed inoltre annunciarne gli aspetti; pertanto di bisogno di SLA.

2.1.2.2 I Service Level Specification

Il termine ***Service Level Specification*** (***SLS***) indica le caratteristiche tecniche di un dato servizio nel contesto di un SLA. Gli SLS contengono le caratteristiche tecniche di un servizio riferite agli aspetti di allocazione del servizio al livello di rete. Gli aspetti non tecnici quali per esempio i metodi di pagamento non sono parte del SLS. Questi sono parte del soprastante SLA. Gli SLS sono parte integrante degli SLA, e inversamente, gli SLA includono un SLS.

In Mescal esistono due tipi di SLS, e quindi di SLA:

- i cSLS, stabilita tra customer e provider
- i pSLS, stabiliti tra due provider

I provider tra cui sono stabiliti due pSLS non devono necessariamente essere interconnessi. Nel caso più generale un provider (nel ruolo di customer) può

stabilire un pSLS con un provider remoto (con il ruolo di provider). Il termine peering provider, indica due provider che sono interconnessi tra loro, e il termine service peering provider indica che i due provider sono connessi ed hanno stabilito un pSLS tra loro due.

Sugli SLS (e quindi sugli SLA) sono permesse le seguenti operazioni:

- creazione di un SLS
- modifica di un SLS
- cancellazione di un SLS

2.1.2.3 Classi di QoS

Gli accordi basati sulla QoS allo stesso tempo riflettono, e hanno bisogno di essere supportati da corrispondenti capacità dei domini su Internet. Da questo si ottiene la definizione fornita da Mescal di classe di QoS (***QC – Quality of Service Class***): "*Una QC indica la capacità di trasferimento per una con QoS di una rete all'interno di un dominio*[2]".

Tale capacità è espressa tramite un insieme di coppie nome-valore in cui gli attributi esprimono diversi indicatori di performance per il trasferimento dei pacchetti come **one-way transit delay**, **packet loss** e **inter-delay variation** (**jitter**), e i loro rispettivi valori. Data la natura statistica dei parametri di performance per il trasferimento dei pacchetti, i corrispondenti attributi possono non essere invarianti; anzi, dovrebbero fare riferimento a specifici intervalli di tempo, che denotano medie mobili, percentili o percentili inversi. In aggiunta il valore dell'attributo non è un valore assoluto, ma solamente relativo, in relazione alla propria QC. In sostanza una QC non è altro che un insieme di attributi (indicatori di performance) con i loro rispettivi valori.

Si deve notare che le QC non sono di per se dei servizi. Il concetto di QC può essere invece avvicinato al concetto di "***Per Domain Behavior***" (***PDB***) presente

[2] Definizione originale: "A QoS-class (QC) denotes a basic network-wide QoS transfer capability of a Provider domain".

nell'architettura DiffServ. Le QC sono associate ad un certo numero di vincoli, i quali rappresentano condizioni relative alla disponibilità di tempo e della topologia. I vincoli sul tempo sono espressi in periodi durante i quali la QC può essere resa più o meno disponibile. I vicoli di topologia sono espressi in termini di raggiungibilità di domini attraverso i quali sia disponibile la QC desiderata.

Considerando un dominio, l'allocazione delle risorse relative ad una QC può soltanto basarsi sulle possibilità di network engineering del dominio stesso, quelle legate al routing e alla gestione delle risorse (banda e buffer). Le quali risultano dalla combinazione delle potenzialità di DiffServ con QoS, insieme con l'utilizzo di funzione di ***Traffic Engineering*** (***TE***). In aggiunta alle proprie abilità un dominio può fare affidamento per il provisioning di una classe di QoS end-to-end anche sulle potenzialità (QC) offerte da altri domini provider.

Le QC supportate da un dominio possono essere distinte in locali (l-QC) ed estese (e-QC):

- l-QC indica la potenzialità di trasferimento con QoS che può essere offerta impiegando solamente i mezzi del provider stesso. Evidentemente i confini di un dominio che appariranno nei vincoli di topologia di un l-QC dovranno appartenere al dominio provider.
- e-QC indica la potenzialità di trasferimento con QoS che può essere offerta impiegando non soltanto i mezzi di un dominio, ma anche utilizzando opportunamente i mezzi (le risorse) messi a disposizione dagli altri domini. In altre parole una e-QC viene offerta utilizzando le potenzialità del dominio provider più le potenzialità appropriate degli altri domini provider. I confini di un dominio che appariranno nei vincoli topologici di una e-QC potranno essere al di fuori del dominio provider, questo grazie all'estensione della visibilità della topologia della QoS del dominio provider.

Per concludere, il termine classe di QoS denota un classe di QoS sia locale che estesa, senza fare particolare rifermato ad una delle due in particolare.

2.1.3 Gli SLS

Attualmente il metodo più utilizzato per offrire un servizio e quello di basarlo su di un contratto. Il termine Service Level Agreement è largamente utilizzato per definire tale accordo o contratto. Esso descrive le caratteristiche del servizio offerto e le responsabilità reciproche delle parti coinvolte nel utilizzo o nell'offerta del servizio. Il termine Service Level Specification è invece utilizzato per indicare le caratteristiche tecniche del servizio offerto all'interno del contesto di un SLA. Le caratteristiche tecniche del servizio si riferiscono agli aspetti di provisioning del servizio e.g. aspetti di richiesta, attivazione e consegna nell'ambito della rete. Gli aspetti non tecnici riguardanti il provisioning come canoni e costi non sono parte degli SLS ma dei degli SLA soprastanti. Gli SLS sono parte integrante degli SLA, e vicendevolmente, gli SLA includono al loro interno gli SLS.

Mescal è focalizzato sugli SLS. L'aspetto contabile ed economico sono infatti al di fuori dello studio fatto dal progetto. La soluzione fornita da Mescal per la consegna con QoS sulla rete Internet adotta un modello hop-by-hop in cascata di interazioni tra provider, sia a livello di rete (IP) che a livello di servizio.

I'SLS può essere visto sotto due lati:

- come accordi tra provider per scambiarsi del traffico con QoS
- come servizio basato su QoS offerto da un provider.

Chiaramente c'è una relazione tra questi due aspetti e ognuno punta i suoi requisiti sul concetto di SLS. Analizzando questi requisiti vengono specificati modelli adeguati.

2.1.3.1 Tipi di SLS e specifica dei requisiti

L'SLS forma la base di un accordo tra provider per lo scambio di traffico su Internet. In buona sostanza l'SLS estende, fino alla fine dello scambio di traffico con QoS, i rispettivi accordi che oggi esistono tra provider nell'Internet Best-Effort per il passaggio di traffico, così possono essere in linea con il contesto specifico

dello scambio di traffico che è implicato dalle particolari relazioni di business mantenute tra provider.

Il modello di business, le relazioni e le colonia finanziarie nell'Internet di oggi di tipologia Best-Effort. La soluzione prodotta da Mescal incontra due business case principali:

- Un caso di business per il provisioning di servizi basati su QoS basati soltanto su garanzie di QoS di tipo Loose[3], con espresso il solo obiettivo della performance, e nessuna garanzia sulla banda.
- Un caso di business per il provisioning di servizi basati su QoS basati su garanzie statistiche per un obiettivo di performance quantitativo, e la banda, in aggiunta alle garanzie di QoS qualitative.

Si può notare che in entrambi i casi sopra elencati anche i servizi basati su garanzie di servizio di tipo Hard[4] possono essere offerti, tramite l'installazione di un tunnel basato su MPLS (***LSP – Label Switched Path***) tra punti specificati della rete. Tuttavia questo servizio non è pensato per il grande mercato, a causa dei suoi limiti di scalabilità presenti nelle soluzioni tecniche.

Il caso del modello di QoS qualitativa corrisponde direttamente al modello gerarchico a tre strati attualmente in uso nell'Internet Best-Effort. In questo caso le relazioni di business tra i provider sono completamente determinate dalla loro posizione relativa nella gerarchia. Si dividono in due tipi:

- Relazioni di tipo customer-provider basate su pSLS, dove l'AS che ha il ruolo del provider offre il servizio di connettività all'altro AS, che ha il ruolo di customer.
- Relazioni di tipo peer-to-peer basate su pSLS, dove i due provider mutuamente si accordano sul traffico di QoS da scambiare tra i rispettivi domini. Questa relazione è una specie di "scorciatoia" per evitare che il

[3] Traduzione: Libere - aperte

[4] Traduzione: Forte - impegnativo

traffico passi nei livelli più alti. Solitamente è utilizzata tra provider di dimensione simile e sullo stesso livello.

Il caso di QoS statistica richiama il modello di un Internet "flat", dove le relazioni di business tra provider non sono soggette alla posizione relativa nella gerarchia a tre livelli. In questo caso sono proposti i seguenti tipi di relazioni tra provider:

- Relazioni con proxy di QoS (upstream) basate su pSLS, dove qualunque dei provider può richiedere all'altro di offrire un accordo di connettività basato su QoS verso ogni destinazione che il secondo provider possa trovare nell'Internet con QoS. Il provider che offre il servizio dovrà avere le proprie capacità di ricerca con QoS basate su accordi simili con alcuni dei provider a lui adiacenti e così via. Quindi ogni provider in una catena di proxy con relazioni di QoS stabilite nella stessa direzione, apparirà come un proxy per i provider successivi lungo quella direzione.

Basandosi sulle precedenti argomentazioni sono stati creati i seguenti tipi di pSLS.

Nel piano di internet gerarchico abbiamo:

- *Provider Loose QoS Internet access pSLS* – utilizzato per relazioni di tipo customer-provider.
- *Provider Loose QoS tunnels in the Internet pSLS* – utilizzato per relazioni di tipo customer-provider.
- *Peer Loose QoS traffic inter-exchange pSLS* – utilizzato per provider in relazioni peer-to-peer.
- *Peer Loose QoS tunnel extension pSLS* – utilizzato per provider in relazioni peer-to-peer.

Nel piano di un Internet di tipo flat abbiamo:

- *Proxy statistically guaranteed QoS Internet access pSLS* – utilizzato per relazioni con proxy di QoS, offerto da provider che vogliono offrire un servizio di transito di QoS.

- *Proxy statistically guaranteed QoS tunnels in the Internet pSLS* – utilizzato per relazioni con proxy di QoS.

I pSLS elencati precedentemente differiscono gli uni dagli altri nel tipo di garanzie di QoS offerta, la direzione, e lo spazio di visibilità geografico.

Nelle relazioni di tipo customer-provider i pSLS assumono una connotazione di accordo per il provider, nel ruolo di customer di "legarsi" alla parte di Internet che utilizza la QoS. Questo implica la bidirezionalità dei flussi del traffico di QoS e può solamente offrire garanzie di servizio qualitative verso tutte le destinazioni che possono essere raggiunte dal provider nel ruolo di provider.

I pSLS tra provider peer-to-peer hanno la connotazione di un accordo reciproco per lo scambio di traffico dal dominio di un provider al dominio dell'altro provider. Questo implica un flusso bidirezionale e offre garanzie del servizio qualitative all'interno dei domini dei provider.

Nella relazione di proxy di QoS, i pSLS hanno la connotazione di accordi per l'offerta di pSLS. Questo implica un flusso unidirezionale diretto dal richiedente al provider, e può offrire garanzie del servizio di tipo statistico e/o qualitativo verso certe destinazioni verso le destinazioni internet raggiungibili dal provider.

Una volta che i pSLS sono stati installati al fine di stabilire tunnel con qualità del servizio i provider interessati possono offrire ai loro end-customer cSLS con garanzie di qualità del servizio di tipo Hard.

Fino ad adesso i pSLS sono stati visti come accordi, sottolineando il loro aspetto di relazioni istaurate tramite provider. Analizzeremo adesso il loro aspetto più intrinseco riguardante le caratteristiche del traffico QoS che essi implicano.

Sono richiesti differenti tipi di pSLS a causa dei diversi tipi di relazioni che possono esistere tra provider; di conseguenza le informazioni specificate nei pSLS devono venire incontro ai seguenti problemi:

- Offrire un linguaggio "well-known", compreso per descrivere il contenuto dei pSLS in maniera che possa soddisfare i seguenti requisiti:

- Catturare gli aspetti fondamentali degli accordi tra provider per lo scambio di traffico con QoS come implicito nella loro relazione, al fine di beneficiare, o comunque facilitare le interazioni tra provider e quindi lo sviluppo di nuovi servizi.
- Considerando che un pSLS esprime i servizi offerti sulla base di QoS, creare una base informativa stabile per costruire la gestione dei servizi e funzioni di ***Traffic Engineering*** (***TE***), per automatizzare l'allocazione delle risorse e migliorare la consegna dei pacchetti.

2.1.3.2 Il modello di SLS

Il modello di SLS utilizzato all'interno di Mescal si può ricondurre al modello usato nel progetto Tequila [TEQU]. Il modello di SLS proposto (***SLS-T – SLS Template***) è considerato il nucleo dei servizi IP basato su QoS. L'SLS-T descrive le caratteristiche tecniche di una singola topologia, flussi di dati IP, caratteristiche di qualità del trasferimento, criteri di conformità del traffico. Gli aspetti di connettività di un servizio quindi sono una raccolta di SLS-T legati al solito customer con gli stessi mezzi di accesso/utilizzo e caratteristiche.

Di seguito viene presentata la struttura del modello di SLS, sottolineando i miglioramenti effettuati nel progetto Mescal.

Il modello di SLS contiene i seguenti elementi, chiamati anche clausole:

- **Identificativo del SLS** – una chiave che identifica univocamente il SLS, decisa dal provider.
- **Visibilità** – identifica la regione geografica, o comunque topologica nella quale la politica di QoS specificata nel SLS è valida. Vengono indicati le estremità di tale regione tramite gli attributi:
- **Ingress** – Punto in cui inizia la validità del SLS
- **Egress** – Punto in cui termina la validità del SLS
- Identificativo del flusso – Definisce il flusso dei datagrammi IP, per i quali, come specificato nell'SLS, devono valere le politiche di QoS. Di

norma, ogni SLS deve avere un unico Identificativo di flusso. Esso include i seguenti attributi:

- **Informazioni su DiffServ** – Indica i possibili valori del campo DSCP nell'header del pacchetto IP per caratterizzare i pacchetti legati all'SLS.
- **Informazioni sulla sorgente** – Specificano i possibili valori dell'indirizzo IP della sorgente del flusso, inserito nell'header del pacchetto IP.
- **Informazioni sulla destinazione** – Specificano i possibili valori dell'indirizzo IP del destinatario del flusso, inserito nell'header del pacchetto IP.
- **Informazioni sull'applicazione** – Specificano i possibili valori del campo Applicazione nell'header del pacchetto IP.
- **Conformità del traffico** – Descrive i criteri, o meglio le caratteristiche , che il traffico che passa per la rete deve rispettare, al fine di mantenere le garanzie di QoS specificate nella clausola Garanzie di performance. Di solito questa clausola contiene informazioni per configurare i condizionatori di traffico ai bordi del provider. Questa clausola contiene i seguenti attributi:
- **Algoritmo di conformità del traffico** – Specifica il tipo di meccanismo utilizzato per identificare in maniera non ambigua i pacchetti che sono conformi al profilo del traffico.
- **Parametri di conformità del traffico** – Un insieme di parametri richiesti in input dall'algoritmo di conformità del traffico.
- **Trattamento degli eccessi** – Questa clausola descrive come viene gestito il traffico che non è conforme con, o meglio eccede, il profilo di traffico presente nell'SLS. Questo processo prende atto dopo ce il flusso è stato analizzato dall'algoritmo di conformità del traffico. La clausola include i seguenti parametri:
- **Azione** – Indica l'azione compiuta sul traffico non conforme.

- **Parametri dell'azione** – Un insieme di parametri richiesti dall'azione stessa.
- **Garanzie di performance** – Questa clausola descrive le garanzie sui parametri delle performance di trasferimento dei pacchetti (metriche) che il provider offre ai pacchetti appartenenti al un particolare SLS, all'interno del limiti topologici dell'SLS (clausola Visibilità). Le garanzie che sono date sono soggette ai criteri di conformità del traffico (clausola Conformità del traffico). Le garanzie sono date per ogni livello di conformità; In caso di più livelli di Algoritmo di conformità del traffico. Per il traffico che non rispetta tali criteri non viene fornita alcun tipo di garanzia. Questa clausola comprende i seguenti attributi:
- **Delay** – Misura il ritardo one-way misurato tra il punto di ingresso e il punto di uscita specificati nel SLS.
- **Jitter** – Misura il ritardo one-way misurato tra il punto di ingresso e il punto di uscita specificati nel SLS.
- **Loss** – Specifica le garanzie per la probabilità di perdita dei pacchetti; questa è definita come il rapporto trai pacchetti in profilo persi, e i pacchetti in profilo inseriti nel flusso.
- **Throughput** – Specifica la frequenza con la quale il traffico è consegnato, che è misurato, ad un preciso punto di uscita, contando tutti i pacchetti relativi all'SLS considerato. Si noti che tutti i pacchetti contribuiscono a misurare il throughput, che siano o meno all'interno del profilo.
- **Tipo di accordo** – Specifica il tipo di accordo coinvolto nel particolare SLS. Un SLS può essere richiesto oppure offerto, tenendo conto degli SLA tra provider. Dal punto di vista di ogni parte in un SLA tra provider, un SLS è richiesto per il traffico che attraversa il dominio del provider e un è offerto per il traffico da inviare nel dominio dell'altro provider.

Qui di seguito è riportata la tabella degli elementi presenti negli SLS. La tabella è ripresa dal Deliverable 1.3 di Mescal [MESC].

Element	Description
SLSID	The SLS identification key.
Scope	The topological boundaries of the SLS.
Scope.Ingress/Egress	Directly attached sites may be identified by a postal address or the IPPrefixes/IPAddresses of the customer site given that the access link they are attached to is already known to the system. An IP Address may explicitly point to the boundary link interface. Geographical locations, ISP names and/or IPPrefixes/IPAddresses may point to remote sites, the mapping of which to boundary links is subject to the contracts and routing decisions in place.
FlowID	The characteristics of the flows entitled to use the SLS.
FlowID.ClassInfo	Characterisation of flows based on their class as this is inferred by the ToS/DS-byte of the IP header.
FlowID.SourceIPAddressInfo/ DestinationIPAddressInfo	Characterisation of flows based on source and destination IP addresses.
FlowID.ApplicationInfo	Characterisation of flows based on application information specified by the protocol number, source and destination ports.
TrafficConformance	The conformance criteria that the traffic should meet to enjoy the performance guarantees of the SLS.
TrafficConformance.Algorithm	The traffic conformance algorithm and parameters to apply per conformance rate level.
ExcessTreatment	How traffic not conforming to the conformance criteria should be treated by the provider.
ExcessTreatment.Drop	Denotes that non conforming traffic will be dropped as the default treatment.
ExcessTreatment.Shape	The parameters of the shaper, if shaping is the desired excess treatment.
ExcessTreatment.Remark	The qualitative guarantees desired for non conforming traffic, if remarking is the desired excess treatment.
PerformanceGuarantees	The guarantees offered to conforming traffic in terms of packet transfer characteristics.
OverallTrafficGuarantees	The performance guarantees for the overall traffic, including non conformant traffic in case traffic conformance clause is specified.
TCLevelGuarantees	The performance guarantees for a particular traffic conformance level.
QuantitativeGuarantees	Quantitative guarantees on packet delay (QnDelayGuarantees), loss (QnLossGuarantees), jitter

	(QnJitterGuarantees) and throughput (QnThroughputGuarantees).
QualitativeGuarantees	Qualitative guarantees (e.g. premium/gold) on packet delay (QlDelayGuarantees) and loss (QlDelayGuarantees).
AgreementType	The type of pursued agreement, SLSes are by default requested, however they may also be offered in the context of SSSs between peer providers.

Tabella 1 - Modello del SLS

2.1.4 Servizi basati su QoS e classi di QoS

Quando si applica ad un servizio offerto, il termine Garanzia (di QoS), facciamo riferimento alla garanzia con la quale gli aspetti relativi alla qualità del servizio stesso possono essere offerti dalla prospettiva del provider. Queste aspetti di qualità differenziano i servizi simili tra di loro.

Secondo la filosofia di Mescal, le garanzie di Qualità del Servizio consistono delle seguenti parti:

- Garanzie di performance – Che riflettono la qualità del trasferimento dei datagrammi trasmessi nel contesto del servizio. Considerando che le classi di QoS sono le basi dei servizi di QoS, queste garanzie corrispondono direttamente ai valori dei parametri di performance delle classi di QoS sulle quali il servizio offerto è basato.
- Garanzie di banda – Sono espresse come limite massimo, in unità di banda, del traffico inserito nella rete sulla quale le garanzie di performance sul servizio accordato possono essere offerte.
- Grado del servizio – Denota la probabilità di prendere nella rete richieste del servizio valide.

I tipi di garanzie di QoS dovrebbero riflettersi nei c/pSLS sottolineando l'offerta di servizi basati su QoS. È responsabilità del provider offrire i servizi per assicurare che garanzie sopra citate che possono essere offerte, non siano violate.

2.1.4.1 Il concetto di Meta classe di QoS

La filosofia che sta dietro il modello di meta-classe di QoS (metaQC) sta nel comprendere i bisogni di QoS delle comuni applicazioni. Dovunque gli utenti siano connessi usano più o meno lo stesso tipo di applicazione in contesti simili. Essi inoltre sperimentano le stesse difficoltà a livello di QoS e quindi usano esprimere richieste di QoS piuttosto simili ai loro rispettivi provider. Non ci sono particolari motivi/obiettivi per cui un Provider voglia realizzare un servizio di VoIP con basso delay e basso jitter, mentre un altro Provider voglia realizzarne uno di tipologia differente. Le applicazioni impongono i vincoli alla rete, indipendentemente da dove il servizio sia offerto su Internet. In una rete in cui ogni Provider sceglie le sue particolari QC, senza tenere conto delle scelte fatte degli altri, nasce il bisogno di un concetto di livello più alto di QC, grazie al quale è possibile creare un legame tra QC appartenenti a domini differenti.

Si deve capire quindi che il concetto di meta classe è un concetto astratto. Non è una QC implementata in una rete reale. Una metaQC può essere definita con i seguenti attributi:

- Una lista di servizi per cui la metaQC è particolarmente adatta.
- I limiti per ogni attributo di performance (one-way transit delay, one-way transit variation jitter, loss rate). In aggiunta un valore di priorità può essere assegnato ad ogni attributo di performance della QC.

Solamente un numero limitato di metaQC dovrebbero essere definite. Ogni AS classifica le sue l-QC basandosi su di una metaQC. Una l-QC di un AS può essere collegata solamente con una l-QC di un AS confinante che fa riferimento alla stessa metaQC. Qui di seguito alcune precisazioni sulle metaQC.

- Una metaQC tipica o dominio (anche se questo concetto può essere esteso in maniera lineare).
- Il concetto di metaQC è molto flessibile, nei rispetti di nuove ed impreviste applicazioni.

- Una gerarchia di metaQC può essere definita per un preciso tipo di servizio. Un data l-QC può essere adatta per più di un a metaQC (persino al di fuori della stessa gerarchia). Più l-QC in un dato AS possono essere classificate, e quindi appartenere alla stessa metaQC. Il concetto di ***Per-Domain Behaviour*** (***PHB***) di DiffServ non deve essere confuso con la metaQC. I due concetti condividono caratteristiche comuni nella specifica di alcuni parametri di QoS. Tuttavia i due concetti non sono perfettamente coincidenti. I due differiscono nella loro proposta. L'obiettivo per la definizione di PHB è aiutare l'implementazione di potenzialità di QoS all'interno di un a rete, mentre l'obiettivo di una metaQC è aiutare la negoziazione di un accordo tra Service Provider. Un PHB è pi simile ad una l-QC di una metaQC. In sostanza l'interesse del concetto di metaQC è triplice come elencato in seguito:
    - Offrire linee guida per il collegamento di l-QC.
    - Permettere importanti collegamenti che non hanno conoscenza degli accordi distanti da loro.
    - Rafforzare la coerenza e la consistenza in un AS path con QoS senza conoscere la catena di AS.

2.1.5 Q-BGP

Con lo stato dell'arte attuale la quantità di informazione che deve essere scambiata tra Network Provider tramite protocolli di routine inter-dominio è notevolmente cambiata. Le informazioni di raggiungibilità devono essere più ricche di quanto non siano già quelle degli attuali protocolli, e devono offrire cammini con informazioni utili al fine di facilitare la scelta del cammino migliore al processo di selezione. Tali informazioni possono essere per esempio le informazioni sulla QoS riscontrata sul cammino in questione. Da questo punto di

vista risulta chiaro che i Provider devono migliorare e modificare i protocolli che usano nei loro domini in modo da poter offrire nuovi servizi a valore aggiunto.

Per offrire questi servizi le infrastrutture devono essere aggiornate, in particolar modo le modifiche che devono essere fatte interessano i protocolli di routing e segnalazione già esistenti. I provider devono sviluppare mezzi per trasportare le informazioni di QoS tra i domini in modo che i servizi basati su QoS possano avere una visibilità maggiore e quindi possano essere utilizzati da un maggior numero di utenti.

In Mescal sono state utilizzate due metodologie per la diffusione delle informazioni relative alla QoS.

- La prima soluzione richiede la propagazione di informazioni tramite un solo identificativo deciso durante la negoziazione del pSLS. Caratteristiche di performance addizionali sono negoziate, ma non scambiate a livello di routing.
- La seconda soluzione richiede la propagazione di un insieme di caratteristiche di performance associate ad un identificativo. La natura di tali informazioni relative alla QoS deve essere stabilita durante la fase di negoziazione del pSLS.

Qui di seguito viene descritto come attraverso la caratteristica di ***BGP – Border Gateway Protocol***, che permette di espandere il numero di informazioni trasportate nel sui messaggi, si possa trasportare le informazioni relative alla QoS tra AS adiacenti.

Molti ISP utilizzano il BGP per connettere i propri AS con quelli degli ISP confinanti. Questo è l'unico protocollo di routine inter-dominio che è utilizzato su internet.

I modelli di Mescal si basano sullo scambio di informazioni relative alla QoS tra due AS adiacenti. Questo scambio avviene a livello di servizio e a livello di routing. Consiste nella negoziazione di garanzie di QoS durante la fase di

negoziazione dei pSLS e poi nella loro propagazione (eventualmente parziale). I mezzi per scambiare le informazioni devono avere i seguenti requisiti:

- Devono essere dinamici e scalabili.
- Devono poter propagare i cambi di topologia senza avere un impatto significativo nell'infrastruttura esistente di Best-Effort.

2.1.6 Loose Option

La ***Loose Guarantees Solution Option*** (***LGSO***), o modello Loose, si basa sull'uso sia del protocollo q-BGP, sia sul concetto di metaQC. Il sistema risultante può essere visto come un insieme di metaQC parallele che utilizzato istanze differenti del protocollo di routing inter-dominio.

Quando esiste un accordo (un pSLS), i due domini confinanti si scambiano informazioni di QOS sulla raggiungibilità in termini di adesione a piani relativi a metaQC. Gli accordi influiscono garantendo al dominio remoto di beneficiare delle proprie conoscenze di QoS. Possono essere inoltre negoziate ulteriori politiche. Di conseguenza i messaggi di q-BGP devono includere un identificativo delle metaQC, così ogni messaggio può essere processato nell'ambito del proprio contesto. È necessario inoltre salvare informazioni relative agli update, in ***Routineg Information Base*** (***RIB***) dedicate alla giusta metaQC. Si devono quindi prevedere tante RIB quante sono le metaQC utilizzate e altrettanti processi di decisione.

Quando un pacchetto transita su un insieme di AS, la QoS che sperimenta deve essere consistente, cioè il pacchetto in ogni AS riceve un trattamento appropriato alla sua classe di servizio. Non è detto quindi che i trattamenti saranno tutti uguali, ma sicuramente saranno simili.

Da questo punto di vista, trovare in un annuncio di q-BGP un identificativo di una metaQC può essere sufficiente ad apprendere un nuovo cammino end-to-end. In questo caso però si deve considerare anche le caratteristiche di QoS ottenute dalla concatenazione delle l-QC incontrate lungo il path. Se tali informazioni sono

presenti dentro il messaggio di update allora il decision processo può utilizzarle per scegliere più consapevolmente quello che ritiene il cammino migliore.

2.1.7 Statistical Option

La ***Statistical Guarantees Solution Option*** (***SGSO***), o modello statistico, fa un uso, se così si può dire, opzionale di q-BGP. Ogni pSLS infatti contiene una descrizione esaustiva delle destinazioni raggiungibili, insieme con le relative garanzie di QoS. Le destinazioni sono note in quanto apprese durante la fase di negoziazione dei pSLS; quindi la conoscenza del passo successivo del cammino deriva direttamente dai pSLS. Il sistema di gestione degli AS si occupa di salvare tali informazioni.

Considerando che un protocollo di routing si occupa in primo luogo di trovare rotte, e in secondo luogo, di selezionare le migliori; si può affermare che per la SGSO il q-BGP non fornisce un reale valore aggiunto dal momento che le rotte sono già conosciute e selezionate in precedenza.

Con la SGSO il processo di routing e quello di instradamento si basano su DSCP. Quando q-BGP è attivo tra i due AS adiacenti i suoi messaggi di update devono trasportare, oltre alle informazioni di raggiungibilità, un valore di DSCP che indica al AS upstream il valore DSCP da usare per beneficiare delle garanzie di QoS offerte per la data QC.

2.1.8 Hard Option

La ***Hard Guarantees Solution Option*** (***HGSO***), o modello Hard, sfrutta gli annunci di q-BGP come mezzi per apprendere gli indirizzi IP in domini distanti al fine di costruire dei LSP. La HGSO usa gli annunci di q-BGP per conoscere destinazioni in base alla metaQC; può quindi essere utilizzata contemporaneamente alla LGSO. Esisto infatti due possibili paradigmi di funzionamento:

- Singolo canale di segnalazione – Gli stessi annunci di q-BGP sono utilizzati sia dalla LGSO che dalla HGSO. Il filtro sulle rotte è il solito per

le due soluzioni. Laddove il path non è completamente specificato la LGSO ignora il dettaglio, mentre nella HGSO il PCE tiene conto di questa informazione.

- Doppio canale di segnalazione – Si introduce un meccanismo per distinguere gli annunci di LGSO da quelli di HGSO. Si inserisce negli update di q-BGP un identificativo della soluzione, in questo modo LGSO e LGSO possono essere attivate l'una indipendentemente dall'altra. Le informazioni per la HGSO possono variare secondo due varianti:
    - Annuncio delle terminazioni dei LSP – In questo caso gli annunci differiscono solamente per il valore dell'identificativo della soluzione.
    - Annuncio del solo Path Computation Service Identifier (PCSID) – In questo caso diminuisce il numero di update (uno per ogni AS)

2.1.8.1 Inter PCE Path Communication Protocol

Il livello di garanzie di QoS offerto dagli Internet Network Provider utilizzando soluzioni basate sul TE su semplice IP, non è così soddisfacente per tutti i tipi di servizio, specie per quelli che offrono garanzie molto stringenti. Per questo tipo di utenti le garanzie di QoS sono considerate il requisito più importante. Attualmente questi requisiti possono essere soddisfatti all'interno del singolo dominio, o attraverso una serie di domini, purché gestiti dal solito amministratore

La HGSO fa uso di una entità dedicata chiamata ***Path Computation Element*** (***PCE***), il quale è responsabile, di cercare/calcolare un cammino inter-dominio che soddisfi un insieme di garanzie al fine di poter creare un tunnel inter-dominio con vincoli di QoS detto LSP. Il calcolo di questa parte è distribuito e richiede un protocollo di comunicazione tra i vari PCE. La comunicazione tra PCE è stabilità grazie ad un protocollo chiamato ***Path Communication Protocol*** (***PCP***). Una volta calcolato, il path è al primo Router del cammino, il quale, attraverso il protocollo RSVP-TE, istalla il path proposto dal PCE.

Nella figura seguente [Figura 1] si suppone che ogni dominio supporti un insieme di metaQC implementate da una seria di l-QC. In aggiunta i pSLS relativi alla HGSO devono essere stabiliti tra domini adiacenti, e di conseguenza, anche una sessione q-BGP; inoltre, almeno un PCE deve essere presente in ogni dominio che supporta la HGSO.

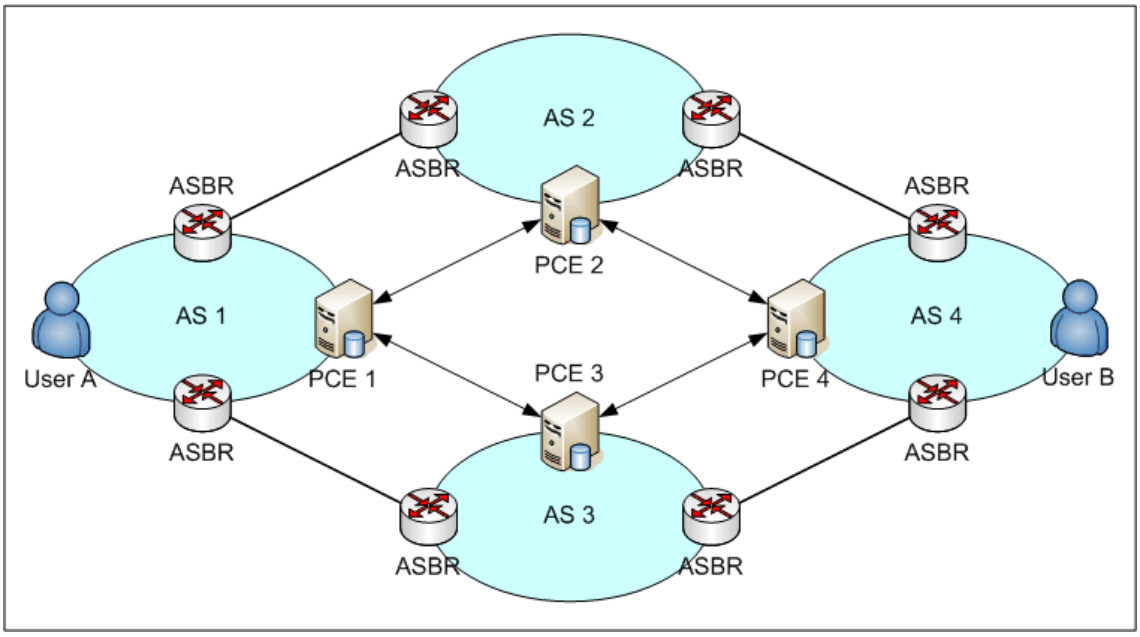

Figura 1 - Scenario HGSO (1)

Per poter calcolare in path inter-AS con garanzie di QoS, il PCE deve poter conoscere in via diretta o indiretta la topologia della rete ed il suo stato all'istante corrente, Per poter accedere a queste informazioni il PCE ha un'interfaccia con il processo di routing; questa interfaccia permette al PCE di conoscere per ogni metaQC, tutti gli AS remoti che supportano la HGSO insieme alle caratteristiche di QoS associate. La figura seguente mostra invece la collocazione del PCE in un AS che supporta la HGSO [Figura 2]

Ogni volta che un pSLS per la HGSO è stabilito, i domini interessati si scambiano le informazioni relative ai propri PCE in modo che essi possano comunicare.

Per creare un LSP inter-dominio con QoS, il dominio a cui interessa la creazione del LSP per prima cosa chiede al suo PCE di calcolare un path inter-dominio che

soddisfi i vincoli di QoS, espressi in termini di disponibilità di metaQC lungo il path, assieme alle garanzie di banda per metaQC e vincoli opzionali quali massimo delay end-to-end ecc. Il primo PCE seleziona uno dei possibili path tra tutti i path candidati ed identifica il dominio successivo. Verifica la disponibilità delle opportune risorse nel proprio dominio e pre-riserva le risorse necessarie. A questo punto contatta il PCE successivo, chiedendo il calcolo di un LSP inter-dominio fino alla medesima destinazione. Questa procedura è ripetuta fino all'ultimo PCE. Se al termine del calcolo è stato trovato un cammino che soddisfa tutti i vincoli richiesti, ogni PCE risponde il path ricevuto, concatenandoci il path interno che esso ha trovato. Quando il primo PCE riceve il risultato il path è disponibile.

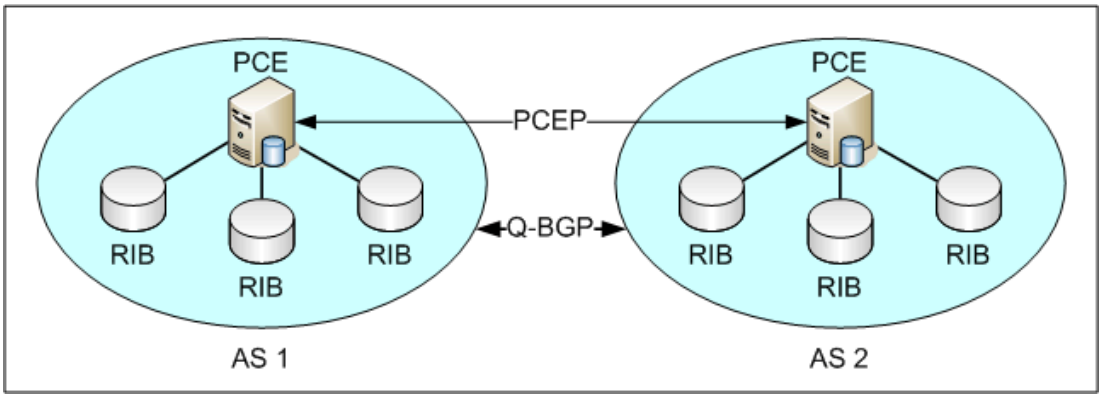

Figura 2 - Scenario HGSO (2)

All'interno di Mescal il PCE è responsabile "solamente" di calcolare un path inter-dominio con garanzie di QoS. Il PCE da solo non installa il LSP inter-dominio, ma calcola semplicemente il path. Un pSLS nell'ambito della HGSO è considerato come un permesso di installare LSP inter-dominio. La destinazione ed il numero di LSP non sono noti a priori, e non sono parte del pSLS. Il pSLS indica solamente i limiti massimi che l'AS a monte può usare, in termini di metaQC che sono usate per stabilire il LSP inter-dominio e in termini di banda massima associata alla metaQC. Il pSLS non riserva nessuna risorsa della rete in anticipo. Le risorse sono allocate quando un LSP inter-dominio è stabilito. Comunque è

difficile stabilire tale contratto in anticipo, specialmente quando il LSP non è noto. Quindi la sequenza di operazioni per stabilire il LSP è la seguente:

- Calcolare i path inter-dominio candidati
- Negoziare le garanzie di QoS inter-dominio lungo il path per il particolare LSP utilizzando le informazioni ricevute dalla computazione del path.
- Stabilire il LSP una volta le garanzie di QoS end-to-end finali sul path sono state concordate.

Stabilire la negoziazione di queste garanzie può essere difficile da realizzare, infatti non è da trascurare la possibilità che le risorse che erano disponibili durante il calcolo del cammino non lo siano in un secondo momento. Per risolvere questo problema è necessario che il PCE di ogni dominio pre-riservi le risorse necessarie e indichi le caratteristiche del path. Questa operazione ha un tempo di validità limitato, oltre il quale le risorse, a meno che non sia avvenuta conferma dell'installazione del path, devono essere rilasciate.

Una volta che il contratto è stato validato, il cammino calcolato dal PCE può essere offerto al primo router del cammino, che effettivamente installerà il path.

Adesso descriveremo il protocollo utilizzato dai PCE per comunicare, e quindi collaborare alla creazione di LSP. Questo protocollo si chiama ***Path Communication Protocol*** (***PCP***) e le principali caratteristiche del protocollo sono:

- Modello di tipo Client/Server, dove un PCE può agire sia da server, che da client (PPC)
- Utilizzo di TCP come protocollo di trasporto, per garantire uno scambio di messaggi affidabile
- Nessun livello di sicurezza per i messaggi per garantire autenticazione, integrità ecc.
- Supporto per un numero limitato di funzioni per il calcolo del LSP.

Il PCP fa uso di nove differenti tipi di messaggi. Qui di seguito verranno elencati i messaggi ed una breve descrizione del loro utilizzo.

- OPEN – Apre una sessione del PCP. Deve essere inviato prima di ogni messaggio scambiato.
- ACCEPT - Si utilizza per rispondere positivamente ad un messaggio di tipo Open. Contiene il tempo di keep-alive per la connessione.
- CLOSE – Si utilizza quando si vuole informare le la corrente connessione non è più disponibile. Il codice di errore spiega il motivo per cui la connessione è stata chiusa.
- REQUEST – Si utilizza quando si deve trovare un potenziale path verso la destinazione scelta. Questo può essere generato anche come conseguenza della ricezione a monte, di un altro messaggio di tipo Request.
- RESPONSE – È inviato in risposta ad un messaggio di tipo Request, ed è inviato quando è stato calcolato un path end-to-end valido. Il messaggio di tipo Response deve partire dall'ultimo dei PCE appartenenti al LSP.
- ACKNOWLEDGE – Questo messaggio è utilizzato per confermare le risorse pre-riservate con il messaggio di tipo Request. Se non viene ricevuto un messaggio di Acknowledge, le risorse allo scadere di un timeout, vengono rilasciate.
- CANCEL – Il messaggio di tipo Cancel è utilizzato per cancellare lo stato della ricerca del path pendente. L'invio di questo messaggio è spesso causato da condizioni di errore.
- PATH-ERROR – Questo messaggio si utilizza per rispondere ad un messaggio di tipo Request; è inviato quando non si trova un path valido oppure non ci sono risorse disponibili.
- KEEPALIVE – Questo messaggio serve semplicemente a mantenere aperta la sessione del protocollo quando non ci sono messaggi scambiati per un periodo di tempo stabilito.

2.2 L'architettura dei PCE in IETF

Il calcolo di Path in reti molto estese e molti dominio è un'operazione complessa e può richiedere particolari componenti e la cooperazione tra gli elementi in domini differenti. In questo capitolo specificheremo l'architettura di del modello di PCE pensato dallo IFTF per risolvere questo problema.

Il PCE è il blocco di partenza per questo tipo di architettura dal quale può essere costruita.

Secondo IETF il PCE viene definito come "un'entità che è capace di calcolare un cammino in una rete, basandosi su grafico di quest'ultima, a di applicarci vincoli computazionali durante il calcolo[5]". Il PCE è di fatto un'applicazione che può essere posizionata all'interno di un nodo nella rete, oppure un server, al di fuori della rete. Per esempio il PCE è capace di calcolare il cammino di un TE LSP operando su un database con informazioni di Traffic Engineering, e considerando banda e altri vincoli applicabili sul TE LSP del servizio richiesto.

Un dominio è un insieme di elementi di rete all'interno una sfera comune di indirizzi o responsabilità di calcolo dei cammini. Esempi di domini includono aree IGP, Autonomous Systems, e più AS all'interno della rete del solito Service Provider. I domini di responsabilità del calcolo dei cammini possono anche esistere come sotto dominio o sotto area o AS.

Al fine di chiarificare esattamente cosa è un PCE si devono prendere in considerazione le seguenti affermazioni:

- Il calcolo di un path può essere fatto intra-dominio, inter-dominio, e in contesti inter-livello.

[5] Definizione originale: A Path Computation Element (PCE) is an entity that is capable of computing a network path or route based on a network graph, and of applying computational constraints during the computation.

- Il calcolo inter-domain può comprendere l'associazione di topologie, protocolli di routing, politiche da più domini dalle quali relazioni può essere dedotto per aiutare nel calcolo del path.
- Il calcolo inter-layer si riferisce all'uso del PCE dove più livelli sono coinvolti e quando l'obiettivo è realizzare il cammino su uno o più livelli, sempre tenendo in considerazione la topologia e le informazioni sulle risorse per i livelli coinvolti.
- In un calcolo di un path un PCE può essere utilizzato per calcolare un path in un dominio. Ci possono essere più PCE in un dominio, ma solo uno di essi è utilizzato nel calcolo di un path.
- Se le richieste sono molteplici allora possono essere utilizzati più PCE.
- Il modello di PCE centralizzato si riferisce ad un modello dove tutti i path in un dominio sono calcolati da un unico PCE.
- Il modello distribuito al contrario indica che il calcolo di un path è separato tra più PCE.
- I cammini che si estendono su più domini possono essere calcolati utilizzando il modello distribuito con uno o più PCE responsabili dei rispettivi domini, oppure con il modello centralizzato definendo un dominio che ne comprende tutti gli altri. Da queste definizioni, un modello di calcolo di tipo centralizzato esegue una solo calcolo di path alla volta; mentre il modello di tipo distribuito può eseguire sia un singolo calcolo di path alla volta, che più calcoli contemporaneamente.
- Il PCE può essere posizionato o meno all'inizio del path. Per esempio, un sistema classico di procedere è quello di avere il calcolo del path all'inizio dell'installazione del LSP con MPLS TE, in questo caso il primo LSR contiene il PCE. Esistono comunque soluzioni in cui altri nodi lungo il cammino possono contribuire alla creazione del LSP. Allo stesso momento possono esistere alcuni casi in cui il PCE è fisicamente distinto dai router del path.

- Il path calcolato dai router può essere un path di tipo "Explicit", cioè con la lista dei passi completa dall'inizio alla fine del path stesso, oppure può essere di tipo "Strict/Loose", che indica la presenza di un insieme di passi di tipo Strict o Loose, che comprendono almeno un passo Loose. Nei cammini Strict/Loose ci possono essere anche passi astratti, in cui un passo indica per esempio un AS anziché un router.
- Il modello di calcolo di un path non significa che sia esclusivo e può essere usato insieme ad altri modelli di calcolo. Per esempio un LSP inter-dominio può essere calcolato utilizzando il modello di PCE in alcuni AS, mentre in altri si possono utilizzare altre tecniche. Infine, per richieste differenti si possono utilizzare tecniche differenti.

Come si può vedere da questi esempi, il PCE non sostituisce il modello esistente di Internet dove l'intelligenza è distribuita all'interno della rete. Al contrario, si basa su questo modello e fa uso di centri di informazione distribuita con abilità computazionali. Il PCE infatti, non dovrebbe essere visto necessariamente come architetture centralizzata, ma come un'operazione cooperativa, o una funzionalità distribuita utilizzata per risolvere problemi specifici come il calcolo del cammino più breve tra più domini.

Ci possono essere più tipi di situazioni in cui il calcolo di un path può richiedere una notevole quantità di risorse, per esempio la creazione di più LSP all'interno di un dominio per ottimizzare una funzione obiettivo, oppure il calcolo di un path attraverso molti criteri (delay, utilizzo del link, capacità di switching, tipi di codifiche ecc.), o ancora il calcolo del costo minimo per la creazione di un albero punto-multi punto.

In questi casi è desiderabile, ma non sempre possibile per alcuni router, che una seconda entità si faccia carico del costo computazionale, a causa delle poche risorse, in termini di CPU, presenti nel router. In questo modo il calcolo del cammino può essere fatto demandando ad un altro router, o ad un server la spesa computazionale.

2.2.1 Architetture dei PCE

In questo capitolo verrà illustrata l'architettura del modello di PCE in IETF.

2.2.1.1 Il PCE come nodo composito

La figura seguente [Figura 3] mostra i componenti tipici del PCE di tipo composito, che è un router che implementa anche le funzionalità del PCE. Il protocollo di routing è utilizzato per scambiare informazioni di Traffic Engineering, sulle quali è basato il ***Traffic Engineering Database*** (***TED***). Le richieste per la creazione di LSP TE sono ricevute dal nodo stesso, e in seguito convertite richieste di segnalazione, ma questa conversione richiede un calcolo del path da parte del PCE. Il PCE lavora sulla TED soggetto a regole locali, al fine di rispondere alla richiesta con il path desiderato.

Si può notare che l'adiacenza a livello di protocollo di routing tra PCE compositi e gli altri router può essere ottenuta tramite mezzi di connessine diretta, o attraverso meccanismi di tunneling.

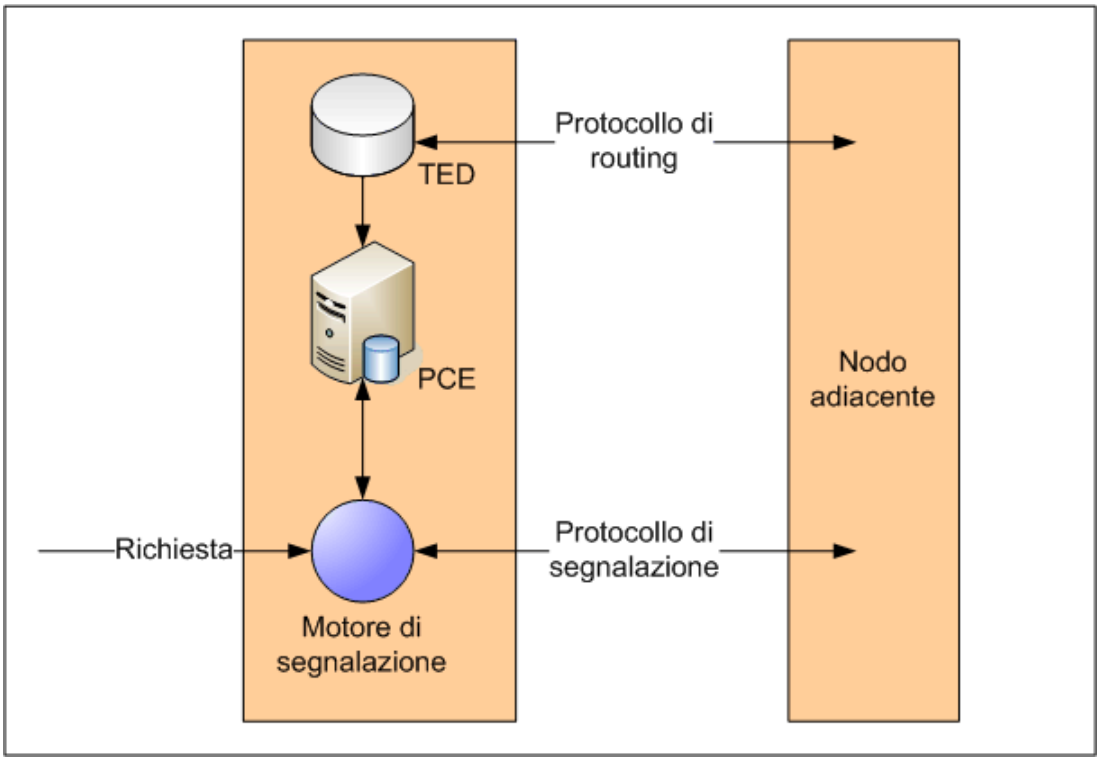

Figura 3 - PCE composito

2.2.1.2 Il PCE come nodo esterno

La figura successiva [Figura 4] mostra un PCE che è esterno all'elemento della rete. Un nodo della rete, quando riceve una richiesta, e prima che inizi la fase di segnalazione, per installare il servizio, esso invia un richiesta di calcolo del path al PCE esterno. Il PCE utilizza la TED soggetta alle regole e politiche amministrative locali e restituisce al nodo la risposta.

Si può notare che il nodo che supporta le funzionalità del PCE può essere anche un router, non è necessario sia una macchina dedicata al servizio di PCE.

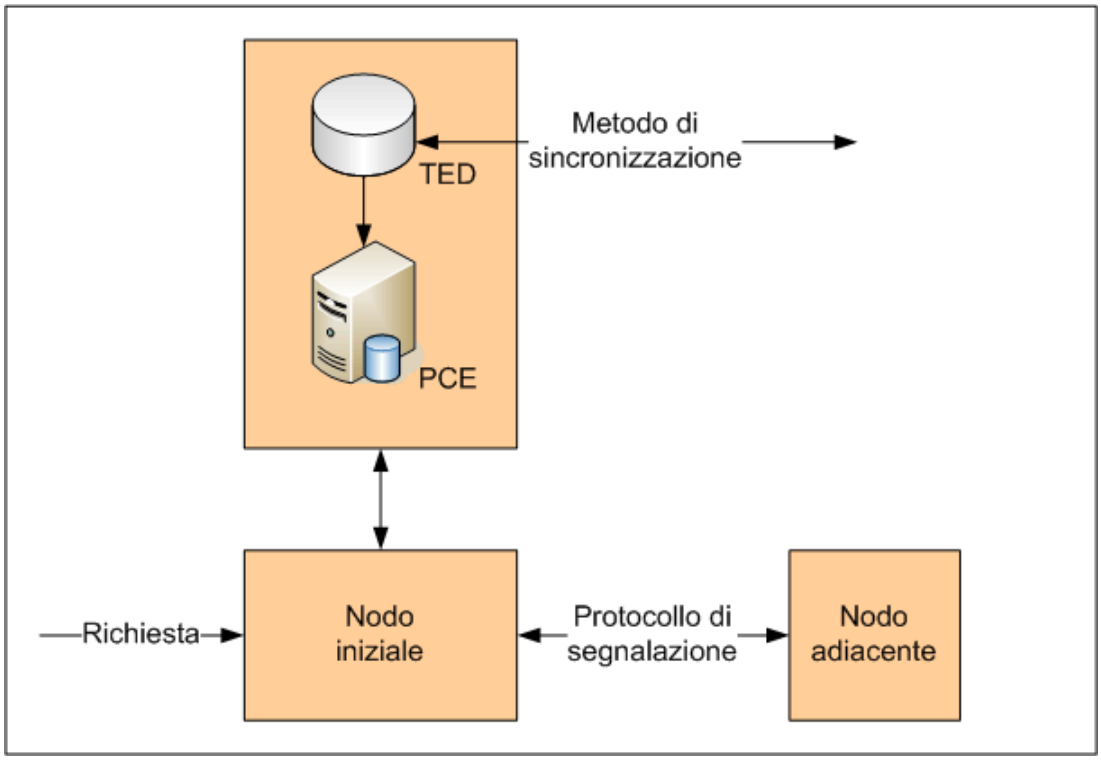

Figura 4 – PCE esterno

2.2.1.3 Calcolo di path Multiplo

Di seguito [Figura 5] è illustrato un esempio in cui lungo il cammino possono essere compiuti più calcoli di path. Come nell'esempio precedente il primo nodo della rete, agendo da ***Path Computation Client*** (***PCC***), invia una richiesta ad un PCE esterno, ma questa volta il PCE restituisce un path che è parziale, ed un PCE lungo il cammino indicato potrà completare. Sarà quindi necessario inviare una (o

più) richiesta per ottenere il path completo. Questo può essere il caso in cui il path non raggiunge la destinazione richiesta, oppure il path restituito è di tipo Loose. I nodi nella rete successivi consulteranno a loro volta altri PCE per ottenere il passo successivo nel path. In questo caso tutte le decisioni riguardo alle politiche utilizzate sono prese indipendentemente da ogni singolo PCE sulla base delle informazioni da esso possedute, e dal quelle fornitegli dal PCC.

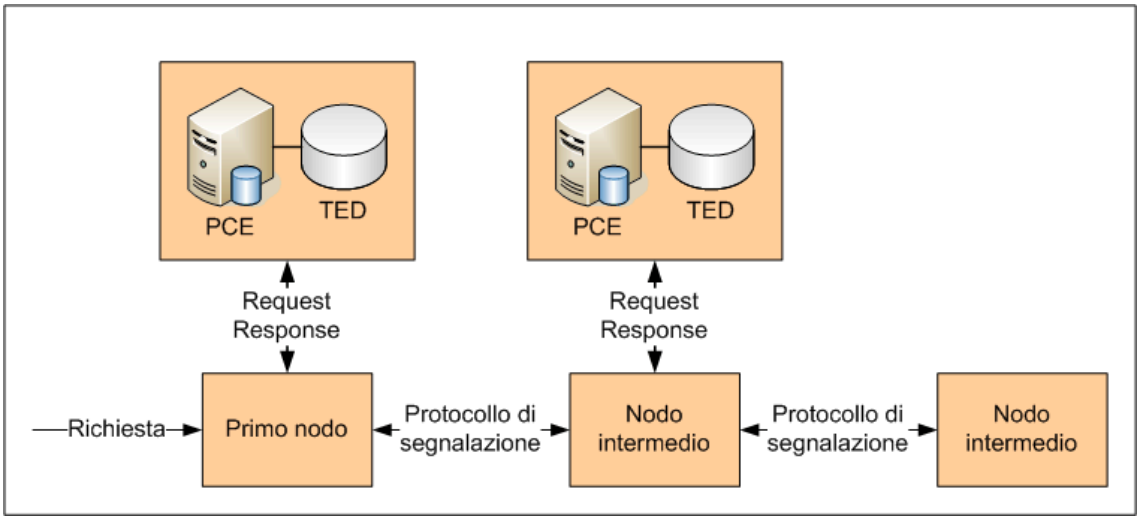

Figura 5 - Calcolo multiplo di un path (1)

Si noti che in questo scenario possono essere presenti entrambi i tipi di PCE, sia il modello composito che il modello esterno.

Nel precedente esempio [Figura 5] il PCE non era in grado di ottenere un path completo per la richiesta che gli era stata fatta. In questo modo i nodi nella rete successivi a quello che ha fatto la richiesta devono preoccuparsi di inviare ulteriori richieste ad altri PCE. Nella figura successiva [Figura 6] viene invece mostrato come possa essere risolto il medesimo problema, introducendo la comunicazione tra i PCE, e facendo si che i PCE cooperino tra loro, così che il PCE interrogato dal primo dei router possa lui stesso interrogare un altro PCE che lo aiuti con la richiesta che gli è stata fatta.

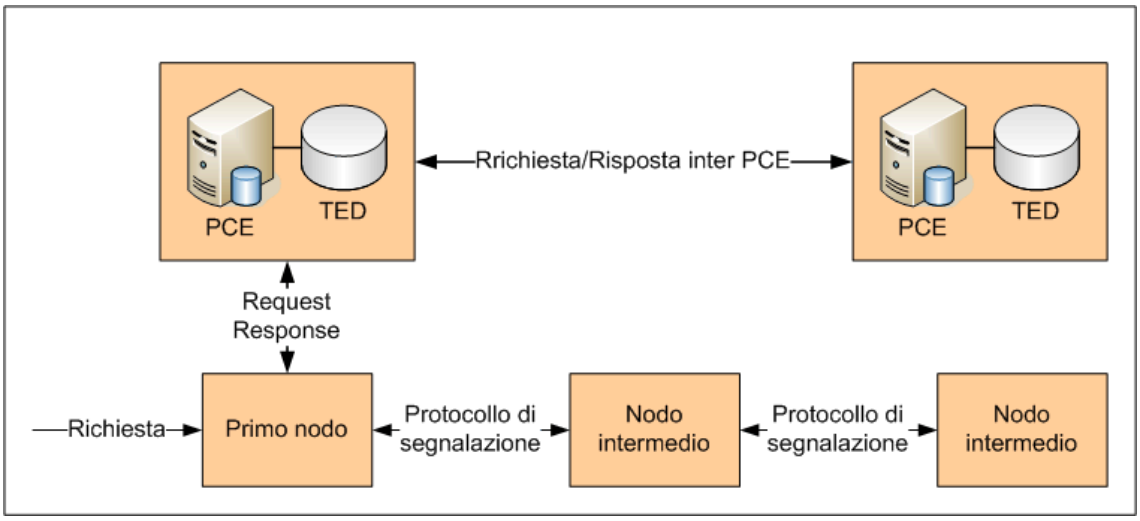

Figura 6 - Calcolo multiplo di un path (2)

Il calcolo di path con integrata la comunicazione tra PCE richiede la coordinazione tra PCE distinti tale che il risultato del calcolo fatto da un PCE dipende dalle informazioni sul frammento di cammino fornite dagli altri PCE. Per questo modello non è previsto un algoritmo di calcolo distribuito, ma permette a PCE distinti di essere responsabili del calcolo di parte del cammino.

2.2.1.4 Il modello centralizzato

Nel modello di calcolo centralizzato si considera che tutte le richieste per un dato dominio siano effettuate da un solo elemento, di tipo centralizzato. Questo può essere un server dedicato , oppure un router designato all'interno della rete. In questo modello tutti i PCC nel dominio devono inviare le loro richieste ad un PCE centrale, mentre un dominio in questo contesto può essere un'area IGP oppure un AS, può anche essere un gruppo di nodi che è definito dalle sue dipendenze su di un PCE.

Questo modello ha un solo punto di fallimento, il PCE stesso. Al fine di evitare questo problema, il modello centralizzato può designare un PCE di backup che può prendere il controllo delle richieste se. Eventualmente, il primo PCE fallisce. Ogni politica presente nel primo PCE deve però essere presente anche nel secondo sebbene le politiche primarie possano essere loro stesse essere soggette a come

siano implementare sul PCE di backup. Da notare il fatto che in ogni momento ci deve essere solo un PCE attivo in un ogni dominio.

2.2.1.5 Il modello distribuito

Il modello di calcolo distribuito si riferisce ad un dominio o ad una rete che può includere più PCE, e dove il calcolo di un path è condiviso tra i vari PCE. Un dato path può essere a turno calcolato da un singolo PCE (Calcolo del path a singolo PCE) oppure da più PCE (calcolo del path con multipli PCE). Un PCC può essere collegato ad un particolare PCE o può essere capace di scegliere liberamente tra vari PCE; il metodo con il quale viene fatta la scelta tra i PCE non è all'interno di questo testo. L'implementazioni delle politiche dovrebbe essere consistente tra i vari PCE che il PCC può contattare.

Spesso il calcolo di un unico path è effettuato completamente da un singolo PCE. Per esempio questo è di solito il caso di MPLS TE all'interno di una singola area IGP dove il LSR di ingresso/PCE composito è responsabile del calcolo del path, o eventualmente, di contattare PCE esterni. Al contrario il calcolo di una solo cammino con più PCE implica che più di un PCE è coinvolto. Un esempio di questo è dove un passo di tipo Loose viene espanso con più LSR/PCE compositi su di un LSP MPLS TE. Un altro esempio è l'utilizzo di più PCE che cooperano tra di loro per calcolare il cammino di un singolo LSP TE su più domini.

2.2.2 Protocollo di comunicazione tra PCE

In questo capitolo si descrive il protocollo di comunicazione tra PCE, detto "PCE Protocol" (PCEP).

L'architettura su cui si basa il PCE è utilizzata per il calcolo di LSP con MPLS TE o nel caso più generale GMPLS TE. Quando un PCC ed un PCE non sono collocati nello sesso apparecchio, è necessario il supporto di un protocollo di comunicazione tra i due, e il PCEP è stato creato appositamente per questo motivo. Un PCC può utilizzare il PCEP per inviare una richiesta per uno o più LSP

TE; e il PCE può rispondere a tale richiesta, sempre con il solito protocollo, se sono stati trovati uno o più path che soddisfacevano i requisiti espressi dal client.

Il PCEP viene instaurato sopra ad una sessione TCP, che garantisce il servizio di affidabilità sei messaggi e controllo del flusso, in modo da non gravare questo compito al PCEP.

Sono definiti diversi tipi di messaggi nel PCEP:

- Open – È utilizzato per iniziare una sessione del protocollo. Questo messaggio è inviato dal PCC al PCE, e in seguito dal PCE al PCC per iniziare una sessione del PCEP. Una volta che è stata stabilita una connessione TCP, il primo messaggio inviato dal PCC al PCE deve essere un messaggio di tipo Open. Ogni messaggio ricevuto prima di questo genera una condizione di errore nel protocollo e quindi la sessione di PCEP deve essere terminata. Il messaggio di Open è utilizzato per stabilire una sessione di PCEP. Durante la creazione della sessione si possono scambiare varie caratteristiche della sessione. Se entrambe le parti si trovano d'accorto su quest'ultime la sessione viene stabilita.
- Keepalive – È utilizzato per mantenere una sessione del protocollo nello stato attivo. Il PCEP possiede un proprio meccanismo ti keepalive, utilizzato per garantire che la sessione sia attiva. Questo richiede di conoscere la frequenza a cui ogni parte del PCEP invia messaggi di questo tipo. Il deadtimer è definito come il periodo di tempo dopo lo scadere del quale la sessione viene dichiarata chiusa se nessun messaggio è stato ricevuto. I messaggi di Keepalive sono utilizzati sia dare un acknowledge ad un messaggio di Open quando le parti hanno trovato un accordo sulle caratteristiche della sessione, e sia per assicurare la validità di una sessione di PCEP.
- PCReq – Questo messaggio è inviato dal PCC al PCE per richiedere il calcolo di un cammino. All'interno di questo messaggio devono essere

presenti due informazioni: il RP[6] e i due End-Point[7]. Se uno di questi due manca il PCE che riceve il messaggio deve rispondere con un messaggio di errore al PCC.

- PCRep – Questo messaggio è inviato dal PCE al PCC in risposta alla richiesta di calcolo di un cammino (PCReq). Un messaggio di tipo PCRep può contenere un insieme di cammini calcolati, se la richiesta è stata soddisfatta, oppure in caso contrario, una risposta negativa. Il messaggio deve contenere almeno un oggetto di tipo RP. Inoltre un messaggio di tipo PCRep può contenere un insieme di cammini calcolati, corrispondenti alle richieste che sono state fatte, oppure può contenere multipli cammini per la stessa richiesta. Se la richiesta ricevuta può essere soddisfatta (il PCE trova uno o più path che soddisfano i vincoli), l'insieme dei cammini trovati è inserito nel messaggio PCRep. Se la richiesta ricevuta non può essere soddisfatta, il messaggio PCRep deve contenere l'informazioni relative al motivo del mancato successo.
- PCNtf – È un messaggio di notifica. Può essere inviato sia da un PCC che da un PCE per informare l'altro nodo del verificarsi di un determinato evento. Il messaggio può contenere anche più di un evento da notificare. Se la notifica fa riferimento ad un particolare evento può essere aggiunto un oggetto RP per indicare a quale evento si fa riferimento. Il messaggio può essere inviato in risposta ad una richiesta, oppure in maniera indipendente dagli altri messaggi.
- PCErr – Messaggio inviato ogni qualvolta si verifica un errore nel protocollo PCEP. Il messaggio può essere inviato in risposta ad una richiesta oppure in maniera indipendente dagli altri messaggi. Nel primo

[6] RP: Request Parameters – Specifica le varie caratteristiche che la richiesta di calcolo del path deve avere.

[7] End-Point – Specifica l'indirizzo IP della sorgente e della destinazione del path per il quale è stata fatta la corrente richiesta. L'indirizzo IP può essere espresso sia in formato IPv4 che IPv6.

caso il messaggio deve contenere un insieme di oggetti RP legati alle richieste pendenti che hanno generato il messaggio corrente. Nel secondo caso non è richiesto nessun oggetto RP. Naturalmente il messaggio di errore deve contenere tutte le informazioni necessarie ad identificare il tipo di errore che è avvenuto.

- Close – Messaggio utilizzato per chiudere una sessione del protocollo PCEP. Una volta che una parte della sessione riceve un messaggio di Close deve cancellare tutte le richieste pendenti e deve chiudere la connessione TCP.

L'insieme dei PCE raggiungibili può essere sia configurato staticamente, che scoperto in maniera dinamica, o eventualmente una combinazione delle due. Il meccanismo utilizzato per rintracciare gli altri PCE (Protocollo di discovering) non viene illustrato in questo documento.

Un PCC può mantenere aperte più sessioni di PCEP contemporaneamente, naturalmente con PCE differenti. Allo stesso modo il PCE può avere varie sessioni di PCEP attive con più PCC. L'unica differenza tra i due interlocutori è che la sessione deve essere iniziata dal client; il PCE, in quanto server, è in attesa di connessioni, ma non si preoccupa di creare sessioni.

3 Il Progetto EuQoS

L'obiettivo del progetto EuQoS consiste in: "*research, integrate, test, validate and demonstrate end-to-end QoS technologies to support advanced QoS-aware applications over multiple, heterogeneous research, scientific and industrial network domains*".

3.1 La Qualità del Servizio

All'interno del progetto EuQoS sono fornite differenti definizioni di QoS:

- Una metrica di performance per un particolare servizio, per esempio error-rate o disponibilità.
- La capacità della rete di offrire dei servizi migliori (includendo banda dedicata, jitter e latenza) a determinati tipologie di traffico.
- Un insieme di soluzioni/meccanismi che offrono la capacità di gestire lo stato di una rete sotto condizioni di stress.

Se si riduce la visibilità alle reti IP, si può parlare allora di QoS come:

- Un insieme di meccanismi per distinguere il servizio offerto ad un insieme di pacchetti o flussi rispetto ad un altro.
- Un insieme di requisiti di servizio da riscontrare nella rete mentre si trasporta un flusso IP.

Detto questo si può pensare a varie definizioni, dipendenti al livello a cui si vogliono applicare

- Applicativo
- Trasporto

- Data Link

Allo stesso modo si possono pensare più definizioni basate sulla tipologia delle applicazioni che ne fanno uso:

- Real-time
- Non Real-time

I parametri di QoS sono utilizzati dalle applicazioni per misurare i loro requisiti in termini di garanzie richieste. La specifica della QoS è differente ad ogni livello del sistema ed è utilizzata per configurare meccanismi di QoS ad ogni livello. Ogni parametro di QoS può essere visto come un tipo di variabile con dei valori massimi e minimi; e questi valori sono soggetti a negoziazione tra i vari livelli del sistema.

3.2 L'architettura end-to-end di EuQoS

L'architettura end-to-end di EuQoS ha due punti di vista; il primo è dal lato dello sviluppo della rete attraverso domini di AS, e l'altro è dal lato del software all'interno del AS. La visione dal punto di vista architetturale è mostrata nella figura seguente [Figura 7], mentre la visione software, sovrapposta alla precedente è mostrata nella figura alla pagina successiva a quella suddetta [Figura 8].

L'approccio dell'architettura di EuQoS alla sfida della gestione del sistema utilizza la metodologia "*Divide et impera*". Questo approccio consiste nel trasformare un problema di grosse dimensioni, che in questo caso è la gestione del sistema, in tanti problemi più piccoli ma più facilmente risolvibili. Se osserviamo il l'architettura di EuQoS in una visione orizzontale (osservando quindi i differenti piani) possiamo notare una chiara e distinta separazione tra il piano di servizio e controllo dal piano di trasferimento dati. Il piano di controllo inoltre è suddiviso ulteriormente in due parti: la prima, indipendente dalla tecnologia utilizzata, il cui modulo di riferimento è il ***Resource Manager*** (***RM***); l'altra, sviluppata in maniera

differente a seconda della tecnologia utilizzata, il cui modulo di riferimento è il *Resource Allocator* (*RA*).

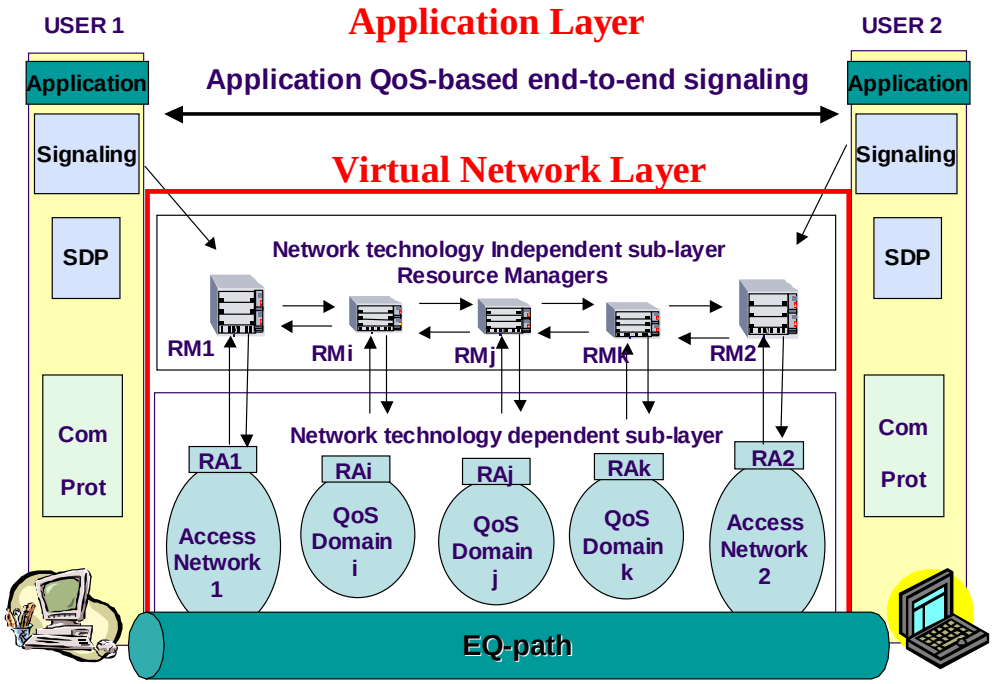

Figura 7 - Architettura di EuQoS (1)

Se l'architettura di EuQoS viene invece vista come divisa in maniera verticale l'approccio "*Divide et impera*" implica una chiara e netta separazione tra le varie tecnologie di accesso e le differenti core network coinvolte nella connessione end-to-end.

Lo scopo dell'architettura di EuQoS non è tanto offrire Qualità del Servizio per tutte le applicazioni, che sarebbe un'impresa difficile da realizzare su larga scala; ma piuttosto offrire QoS solamente per quelle applicazioni che hanno veramente bisogno di tale valore aggiunto. Per questo motivo il sistema EuQoS è basato sul concetto di "sessione".

Quando viene avviata un'applicazione, quest'ultima si preoccupa di avviare la corrispondente fase di inizializzazione lungo la rete per avere il servizio di QoS richiesto. Questo ha vantaggio di sincronizzare perfettamente i requisiti di QoS, il setup e l'utilizzo delle risorse di QoS da parte dell'applicazione. Un problema

ulteriore che viene risolto dal concetto di sessione è il rilascio intelligente delle risorse di QoS al termine del loro utilizzo da parte dell'applicazione. Per questo motivo il sistema EuQoS usa una versione evoluta del protocollo ***SIP – Session Initiation Protocol***, chiamata EQ-SIP, che, in più del predecessore, permette la negoziazione della QoS quando la sessione viene inizializzata.

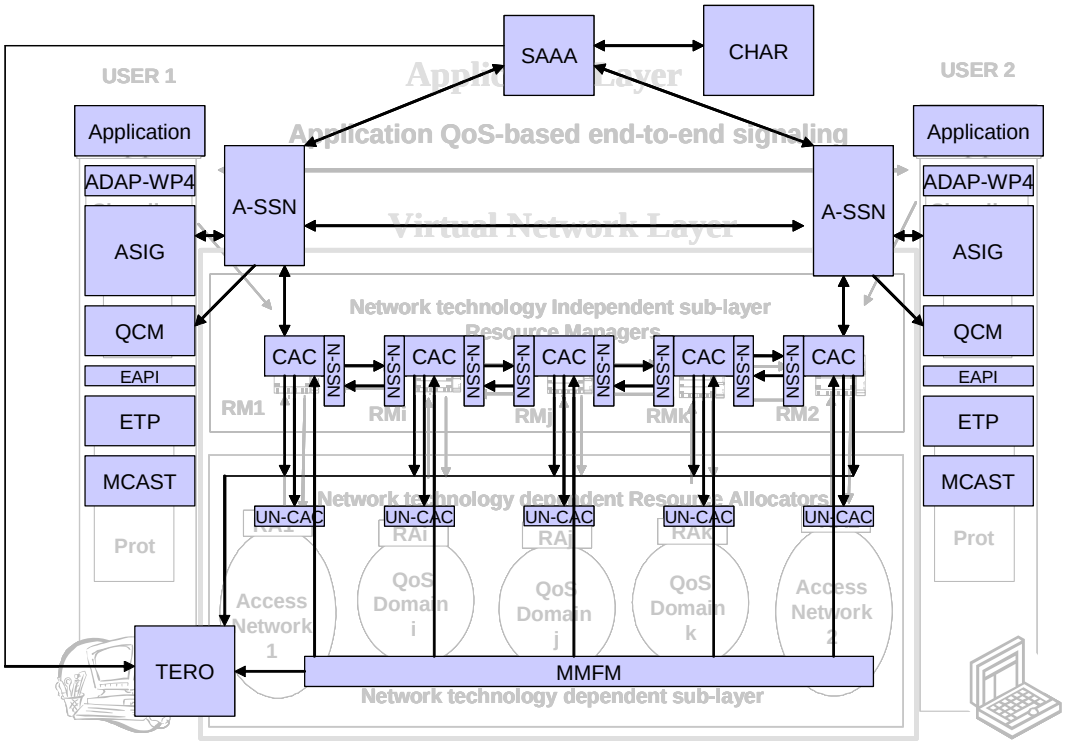

Figura 8 - Architettura di EuQoS (2)

Allo stesso modo, e dopo aver diviso il problema su più livelli e su più parti della rete, il sistema EuQoS utilizza lo stesso paradigma anche nel tempo dividendo le operazioni in tre istanti temporali:

- **Processo di allocazione** – La cui funzione è offrire risorse lungo i differenti AS.
- **Processo di invocazione** – Il cui ruolo è essenzialmente effettuare l'admission control (CAC).
- **Processo OAM** (**Operating and Maintenance**) – La cui funzione è misurare e monitorare il sistema EuQoS. Questo processo offre anche il sotto sistema di fault management.

3.2.1 La definizione dell'EQ-path

La finalità del sistema EuQoS è costruire, utilizzare e monitorare cammini end-to-end con QoS (EQ-path), tenendo in considerazione le garanzie di QoS attraverso i vari AS. Lo scopo di questi EQ-path è offrire un supporto di garanzie di QoS alle applicazioni su un cammino end-to-end. Ogni EQ-path corrisponde ad un dato insieme di parametri di QoS e, in particolare, la Classe del Servizio (***CoS*** – ***Class of Service***).

Se il numero di CoS nel sistema è sotto il controllo dell'operatore, sembra logico pensare che un piccolo numero di CoS sia sufficiente per uno spettro molto più ampio di tipologie di applicazioni.

Le proprietà degli EQ-path devono soddisfare sia le garanzie di QoS, sia i requisiti di scalabilità. Per soddisfare il primo requisito, sarebbe opportuno creare un LSP per ogni flusso di dati, ma questo farebbe si di rendere l'approccio non scalabile. Per soddisfare il vincolo di scalabilità invece si potrebbe adottare l'approccio contrario, cioè costruire grossi "tranci" di traffico per trasportare un largo numero di flussi; purtroppo però questo approccio non garantisce nessuna garanzia QoS per un dato flusso. Il concetto di EQ-path combina i vantaggi dei due approcci, escludendone tutti gli svantaggi.

Gli EQ-path seguono i requisiti del sistema EuQoS. I path end-to-end che esistono nel piano di trasporto, sono costruiti e monitorati dal piano di controllo, per supportare le QoS desiderate dal livello di servizio. Allo stesso modo gli EQ-path sono costruiti tra una vasta quantità di tecnologie e reti. Il sistema EuQoS è responsabile della scelta e dell'unione di ogni parte della rete per formare l'EQ-path.

3.2.2 L'architettura del Resource Manager

Data una visione generale del progetto EuQoS adesso inizieremo ad entrare nel dettaglio software dell'architettura. In questo breve paragrafo viene illustrata la struttura del Resource Manager, il modulo software del sistema indipendente

dalla tecnologia utilizzata. La visione del Resource Manager è fornita per far comprendere meglio l'architettura che sta attorno al modulo sviluppato all'interno del lavoro svolto nel periodo di tesi, in quanto questo modulo è situato all'intero del Resource Manager.

Il Resource Manager è formato da vari sottoblocchi connessi tra loro e da un database a cui tutti i blocchi fanno riferimento. Il database, chiamato RM-DB salva tutte le informazioni dai moduli che compongono il RM. Questo modulo include i vari SLS manipolati dal RM, le informazioni sulla topologia e sulla politica adottata.

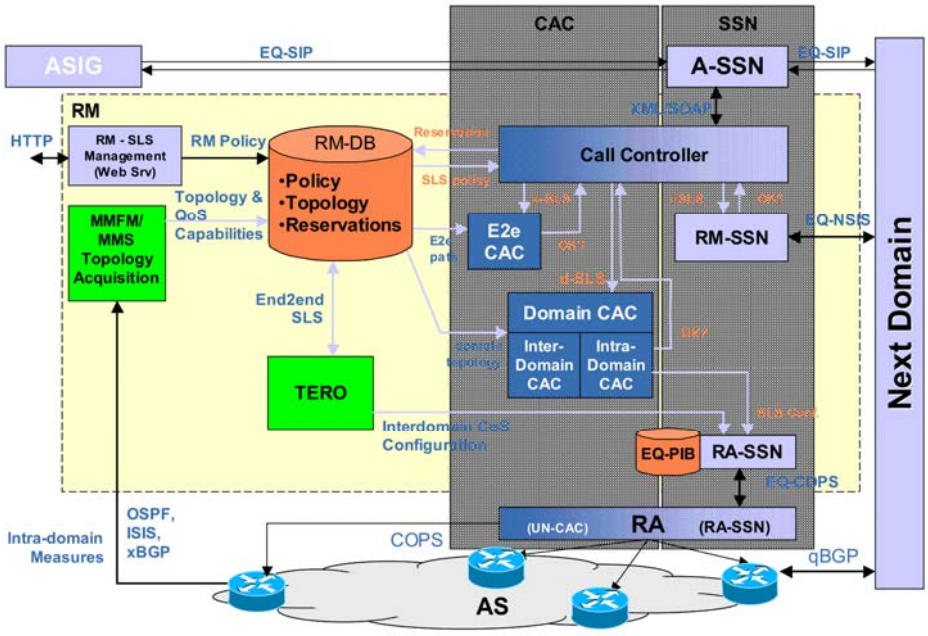

Figura 9 - Architettura del Resource Manager

Come mostrato nella figura precedente [Figura 9], il RM è formato da sei sotto-moduli: tre legati alla QoS:

- ***CAC – Call Admission Control***
- ***TERO – Traffic Engineering and Route Optimization***
- ***MMFM – Monitoring, Measurement and Fault Management***

Due moduli sono legati alla segnalazione:

- ***RM-SSN*** – che gestisce la segnalazione tra differenti RM

- ***RA-SSN*** – che gestisce la segnalazione con il RA

Un modulo legato al database utilizzato:

- ***RM-DB interface*** – che offre i metodi di accesso al database condiviso.

Nelle pagine successive daremo una breve descrizione del Modulo TERO, in quanto "scenario" del progetto legato alla tesi svolta. Per descrizioni riguardanti gli altri moduli si rimanda alla bibliografia [EUQS].

3.2.3 TERO

Per costruire un cammino end-to-end con QoS dobbiamo selezionare un adeguato insieme di parametri ***QoS_NLRI – QoS-related Network Reachability Information*** per definire tale EQ-path. In particolare tali valori del parametro QoS_NLRI non possono essere troppo limitati da AS lungo il tragitto con poche risorse a disposizione. L'utilizzo di questi parametri QoS_NLRI è difficile e può essere strettamente collegato al successo o all'insuccesso del sistema EuQoS.

Il modulo TERO, situato all'interno del RM, costruisce gli EQ-path durante il processo di allocazione; controllando e ottimizzando il processo di routing.

Al livello indipendente dalla tecnologia utilizzata, le rotte di traffico sono identificate tra domini, con l'obiettivo di ottimizzare le risorse inter-dominio basandosi sui requisiti di QoS. Durante la fase di allocazione TERO lavora in backgroung, con il rispetto del sistema EuQoS. Nello svolgere questo compito TERO interagisce con i router di bordo. Più specificatamente TERO:

- Configura il protocollo EQ-BGP per:
    - Tenersi aggiornato tramite le informazioni del NLRI presenti nei messaggi di update ricevuti, tenendo in considerazione i contributi di QoS del dominio.
    - Permettere alle informazioni di NLRI con QoS di essere propagate attraverso i confini del dominio, in accordo ai pSLS negoziati.

- Guidare il motore decisionale di EQ-BGP nel caso di più update per la stessa destinazione, basate su parametri di LOCAL PREF di QoS.
- Configura le code e le modella per offrire le risorse necessarie per permettere al traffico di attraversare i domini confinanti basandosi sui pSLS negoziati.

Per svolgere questo task TERO deve essere a conoscenza di:

- pSLS stabiliti dall'AS
- Link inter-dominio dell'AS
- Traffico all'interno dell'AS
- Valori dei parametri di QoS relativi al traffico all'interno dell'AS.

3.3 Il modello Hard in EuQoS

Per garantire La QoS end-to-end, il progetto EuQoS divide l'intero controllo della QoS stessa in un insieme di più controlli effettuati a livello di dominio. Questo significa che lo sviluppo del path end-to-end con QoS, costituisce il blocco base dell'architettura di EuQoS. Tale path tra entità remote deve essere controllato per assicurare le sue caratteristiche di QoS. Una delle responsabilità principali del sistema EuQoS è rendere le varie tecnologie sottostanti, le quali sono giustapposte al fine di determinare un path end-to-end, cooperanti al fine di garantire garanzie di QoS end-to-end.

Il processi di allocazione in EuQoS è responsabile di stabilire i path end-to-end attraverso i confini dei domini e di offrire una adeguata quantità di risorse lungo il path stabilito per offrire QoS. Per fare ciò è stato creato il modello "Loose" utilizzato nella prima parte del progetto.

3.3.1 Limitazioni del modello Loose

Il modello Loose sviluppa il path di trasferimento partendo dal cammino che seguono i dati: Il camino dei dati è prima selezionato dal protocollo di routing, e in un secondo momento il protocollo di segnalazione si preoccupa di riservare le risorse per in cammino considerato.

Nel modello Loose le risorse sono allocate per Classe di Servizio in ogni AS, indipendentemente dai altri. Dopo il processo di allocazione l'esistenza di un EQ-path garantisce la continuità di un cammino per una data CoS. In ogni caso le risorse necessarie per stabilire una connessione per un singolo utente lungo l'EQ-path sono composte dinamicamente e associate al quel path tramite la funzione di ***Call Admission Control*** (***CAC***) al momento della creazione.

Come esempio di ciò assumiamo che una Classe di Servizio X, la quale richiede una certa quantità di risorse deve essere stabilita tra due end-point. Assumiamo inoltre che l'EQ-path che connette i due end-point sia conosciuto, e attraversi più di un solo AS. Ogni AS attraversato possiede risorse per la CoS X ed esiste un pSLS tra ogni coppia di AS lungo il cammino end-to-end. Secondo il modello Loose, non c'è la possibilità di sapere alla sorgente che ci siano sufficienti risorse disponibili che supportino la connessione. Quindi c'è il bisogno di un meccanismo, in ogni AS attraversato di:

- Verificare che sia disponibile un certo quantitativo di risorse
- Riservare tali risorse

Questa operazione è svolta dal CAC in ogni AS.

In ogni caso gli EQ-path sono stabiliti, nel modello Loose, secondo l'approccio classico del routing inter-dominio. Si utilizza però una versione modificata di BGP, chiamata EQ-BGP per poter includere informazioni di QoS. Più precisamente EQ-BGP avverte la raggiungibilità di una destinazione per una classe di servizio X. EQ-BGP non trasporta però informazioni riguardanti lo stato delle risorse.

Il maggiore vantaggio del modello Loose è che richiede una coordinazione minima tra gli AS lungo differenti EQ-path. Infatti esso richiede solamente accordi

tra AS confinanti al fine di garantire la continuità della CoS. Dall'altro lato il difetto maggiore del modello è quello della scalabilità: il sistema EuQoS non scala sufficientemente bene a causa dell'immane quantità di traffico che genera la fase di segnalazione.

In aggiunta EQ-BGP, in quanto derivante da BGP, è un path vector protocol e come tale, porta con se i seguenti difetti:

- BGP raggruppa le informazioni sulla topologia, in modo tale che un router che riceve un update viene a conoscenza solamente dell'insieme di AS attraversati per raggiungere la destinazione. Inoltre in EQ-BGP anche le metriche per raggiungere la destinazione sono combinate in modo tale da non poter stabilire quale sia l'apporto di ogni AS a tale metrica. Non c'è quindi la possibilità di ottimizzare le risorse lungo il path end-to-end.

Sia BGP che EQ-BGP selezionano il cammino migliore tra tutti i possibili, ed avvertono solamente quel cammino. Il criterio secondo il quale il cammino migliore è scelto è vario e può contenere anche vincoli amministrativi. Nel caso di EQ-BGP tra le metriche tenute in considerazione c'è anche la QoS. In ogni caso questo comportamento limita la possibilità di esplorare cammini multipli tra sorgente e destinazione tramite tecniche di Traffic Engineering.

In aggiunta all'interno di un AS non c'è nel caso generale alcuna garanzia che un path passi per il "miglior" router di bordo quando esistono più collegamenti tra due AS.

3.3.2 Motivazioni per il modello Hard

La motivazione principale dell'utilizzo di quello che è chiamato modello Hard è la limitazione del modello Loose sopra descritta che permette, durante la fase di allocazione, di legare tra loro risorse allocate a cammini che interessano più di un singolo AS. Per risolvere questo problema MPLS TE è un buon candidato dal momento che possiede la maggior parte delle caratteristiche che sono richieste ed

è un forte supporto al business. Con il modello Hard, diventa possibile guardare ad una sequenza di passi, eventualmente in AS differente da attraversare da parte di un EQ-path come un singolo link virtuale. E di conseguenza utilizzare il CAC, al fine di controllare la disponibilità di risorse per una data connessione, durante il processo di invocazione una volta soltanto per l'intera sequenza di passi e AS coinvolti. Tutto questo riduce drasticamente la fase di segnalazione durante il processo di invocazione, specialmente all'interno della core-network, dove ci si aspetta siano gestite molte connessioni end-to-end, pertanto migliora la scalabilità del sistema.

In aggiunta il modello Hard permette di calcolare AS path tenendo in considerazione la disponibilità di risorse e i costi, in aggiunta ovviamente alla raggiungibilità (fornita da BGP) con le metriche di QoS (fornite da EQ-BGP).

Per permettere ciò esso fa uso dei miglioramenti portati ai tradizionali protocolli di routing intra-dominio quali OSPF e IS-IS con le funzionalità di Traffic Engineering, cioè OSPF-TE e IS-IS-TE. Questo permette per esempio un partizionamento effettivo dei costi relativi alle risorse riservate tra i differenti AS e all'interno degli AS lungo il path dalla sorgente alla destinazione, quindi il livello desiderato di QoS end-to-end viene offerto con il minimo ammontare di risorse occupate.

Infine il modello Hard imposta una struttura di supporto per provider per poter collaborare nell'offerta di servizi ***VPN – Virtual Private Network*** con QoS, i quali vanno al di là di un singolo dominio. In questo modo le VPN inter-provider saranno supportate per mezzo dei tunnel inter-provider MPLS-TE.

In conclusione, le motivazioni per implementare il modello Hard possono essere riassunte nel modo seguente:

- Migliorare la scalabilità.
    - Spostando la fase in cui si riservano le risorse lungo l'EQ-path dal processo di invocazione al processo di allocazione.
    - Riducendo la segnalazione durante il processo di invocazione.

- Migliorare il supporto alla QoS.
    - Ottimizzando l'allocazione delle risorse sulla base degli EQ-path
- Estendere il modello del servizio EuQoS
    - Venendo incontro i requisiti specifici di una classe di utenti, tramite l'offerta di servizi VPN inter-provider con una specifica QoS.

Dall'altro lato, dati i vantaggi forniti dal modello Loose, il modello Hard dovrebbe essere considerato come un approccio addizionale per implementare il processo di allocazione di EuQoS, il quale deve essere integrato con il modello Loose. Onestamente MPLS-TE non è disponibile in tutte le tecnologie di accesso supportate da EuQoS, per esempio in UMTS e in WiFi. Quindi il modello Loose può essere considerato una soluzione base per accedere alle reti quando MPLS non sia disponibile; laddove il modello Hard è un promettente miglioramento per le reti che lo supportano. Perciò uno dei più grandi problemi nella definizione del modello Hard è legato alla possibile coesistenza con il modello Loose in maniera consistente ed effettiva.

3.3.3 Specifica del modello Hard

Come precedentemente specificato il motivo dello sviluppo del modello Hard è permettere durante il processo di allocazione di legare le risorse riservate ad un dato path inter-dominio. Ci riferiamo a quest'ultimo come EQ-link. Un EQ-link è visto come un link con caratteristiche di QoS conosciute tra due nodi qualunque nella rete; in quanto tale esso è associato ad una specifica CoS e non ad una sessione, e le risorse sono esplicitamente riservate per il suo uso esclusivo. In pratica un EQ-link viene creato come un tunnel MPLS-TE che può attraversare più domini/AS.

Nell'esempio riportato nella figura successiva [Figura 10], AS1 e AS4 cooperano per stabilire un EQ-link tra R1.2 e R4.1. Questo EQ-link è fisicamente implementato come un tunnel MPLS-TE che attraversa AS2 e AS3 con una banda

riservata allocata. Quando un host nella rete A vuole aprire una connessione con CoS X verso una rete B, è possibile utilizzare l'EQ-link come un link virtuale che connette R1.2 ad R4.1. In questo caso, non è richiesto l'utilizzo del CAC durante il processo di invocazione nell'AS2 e nell'AS3. Infatti gli unici controlli di ammissione che devono essere effettuati sono quelli locali ad AS1 e ad AS4 e quello collegato al transito dell'EQ-link.

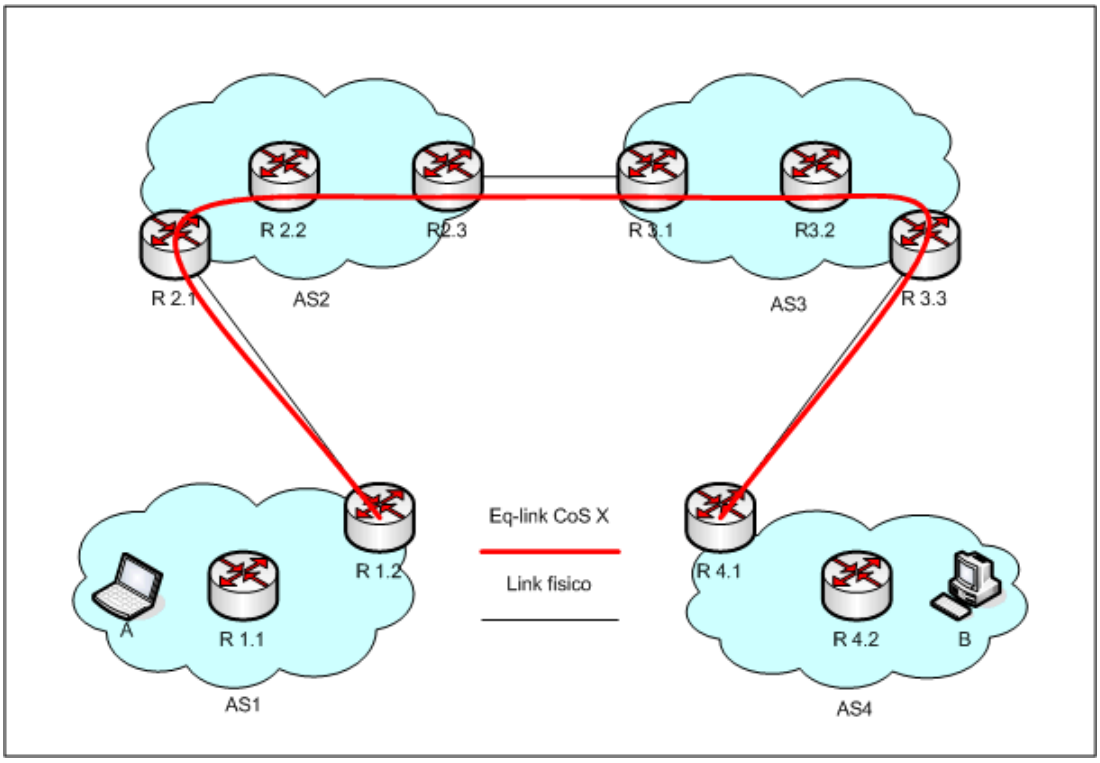

Figura 10 - Esempio di EQ-link

Infine AS2 e AS3 non hanno bisogno di essere a conoscenza della chiamata che avviene tra A e B, dal momento che le risorse sono già state pre-allocate per l'EQ-link sul quale transita la chiamata stessa.

In generale l'EQ-path tra due entità può includere uno o più EQ-link in una riga, la quale può essere o meno contigua. In questo caso è compito del processo di invocazione riservare le risorse sull'EQ-path.

L'obiettivo del modello Hard è quindi quello di realizzare strumenti standard (procedure e protocolli) per creare un EQ-link in maniera consistente come parte del processo di allocazione.

3.4 Requisiti funzionali

3.4.1 Creazione dell'EQ-link

In generale l'EQ-link può attraversare più AS. Esistono tre meccanismi, presi in considerazione da IETF per creare tunnel MPLS che superano i confini di un dominio:

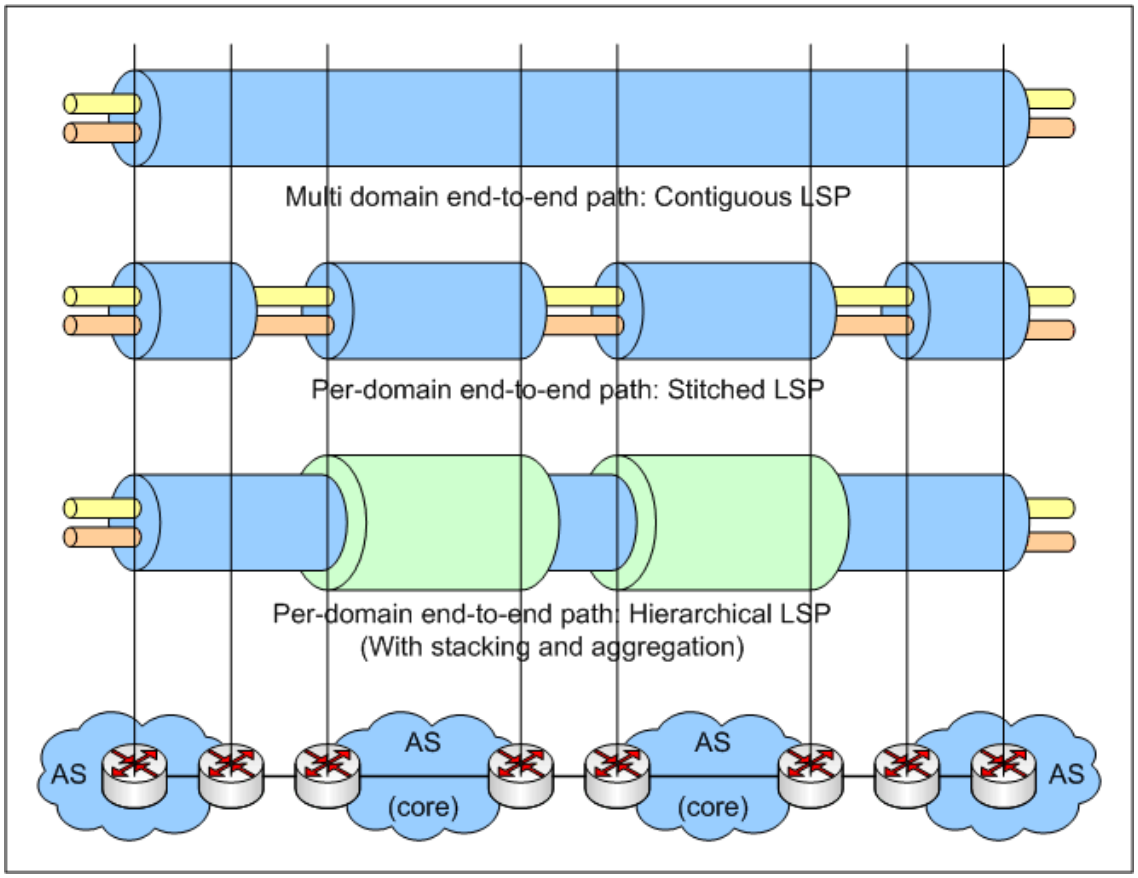

Figura 11 - Tipologie di LSP

- **Contiguous LSP**: Una singola sessione di RSVP-TE con etichette distinte è utilizzata per stabilire un EQ-link. Questo richiede che la sessione RSVP-TE oltrepassi i confini dell'AS, fatto che può creare problemi di sicurezza e inoltre non si attiene alla politica amministrativa dell'operatore. Dall'altro lato questo richiede tanti LSP quanti sono gli EQ-link all'interno dei router nei domini di transito, e ciò non è molto scalabile se il numero di AS è grande.

- **Stitching LSP**: l'esempio è simile al precedente, ma i LSP sono separati lungo il cammino dalla sorgente alla destinazione, e ognuno attraversa un singolo dominio. In questo caso ogni operatore è libero di utilizzare le proprie sessioni di RSVP-TE. Inoltre in questo modo si limitano i problemi di sicurezza, e ci si avvicina alle politiche dell'operatore. Comunque anche in questo caso si richiedono complesse operazioni di classificazione del traffico all'ingresso di ogni AS, e non si risolve il problema della scalabilità delle soluzione precedente.
- **LSP Hierarchy**: Ogni AS ha i suoi LSP, nei quali sono trasportati gli EQ-link (che a loro volta sono dei LSP). Con questo metodo è possibile raggruppare differenti LSP provenienti da differenti sorgenti, ma diretti alla stessa destinazione, in un unico LSP di livello superiore. Da un lato questo lascia l'operatore libero installare e gestire i propri LSP, dall'altro è possibile installare un numero arbitrario di EQ-link all'interno di un singolo LSP, riducendo in questo modo considerevolmente la complessità della gestione in un dominio di transito.

I primi due metodi hanno forti svantaggi per quanto riguarda i problemi di scalabilità; infatti non permettono ad un dominio di transito di trattare il traffico appartenente a differenti EQ-link come un aggregato per quanto riguarda l'allocazione delle risorse. Inoltre il metodo contiguo impone agli operatori di cooperare intensamente per creare un LSP, che inoltre risulta avere problemi di sicurezza dovuti al fatto che la segnalazione per MPLS deve essere aperta all'inter-dominio. Per l'implementazione del modello Hard è stata quindi scelta la terza soluzione. È richiesto un nuovo livello di Admission Control, questa volta nel processo di allocazione.

3.4.2 EQ-link ed EQ-path

Dalla definizione di EQ-link ed EQ-path, possiamo affermare che un EQ-path consiste nella concatenazione di zero o più EQ-link più zero o più path di tipo

Loose (link intra-dominio e inter-dominio standard). Come caso particolare un EQ-path può essere rappresentato da un singolo EQ-link che connette due reti di accesso.

Infine possiamo chiamare EQ-path un cammino di tipo Loose quando esso non include EQ-link.

3.4.3 Supporto per la QoS

La struttura di base per la QoS definita dal progetto EuQoS abilita il supporto della QoS a livello di rete. Tale struttura è situata sopra un architettura di tipo Differentiated Services (DiffServ) nella core network. DiffServ permette di rendere scalabile lo sviluppo della rete con molte classi di servizio. Dal momento che il metodo di instradamento dei pacchetti sugli EQ-link è basato su MPLS, è necessario assicurare che DiffServ sia supportato nei domini MPLS. Per fare ciò deve essere predisposto un meccanismo appropriato che assicuri che ogni classe di servizio riceva un trattamento conforme al suo ***PHB – Per Hop Behavior*** ad ogni ***LSR – Label Switched Router*** nel LSP.

Dal momento che il DSCP è trasportato all'interno dell'header del pacchetto IP, e i router svolgono le operazioni di instradamento basandosi sull'header di MPLS solamente, il problema che sorge è come conoscere l'appropriato PHB a partire dall'header del pacchetto. A seconda della soluzione adottata si può avere:

- PHB della classe di servizio nel campo Exp (E-LSP)
- PHB della classe di servizio presente nell'etichetta (L-LSP)

Nel primo caso uno specifico DSCP è equivalente ad una particolare dei tre bit del campo EXP presene nell'header MPLS, e corrisponde ad una specifica classe di servizio. E-LSP possono trasportare traffico proveniente da ogni PHB, ma in un dominio MPLS possono essere supportati solo otto tipi di PHB. Inoltre non è richiesta nessuna segnalazione per ricevere le informazioni riguardanti il PHB, dal momento che la corrispondenza PHB-EXP è indipendente dall'etichetta.

Nel secondo caso con L-LSP il valore dell'etichetta determina il comportamento dello scheduler, mentre il campo EXP determina la probabilità di perdita assegnata al pacchetto. Inoltre L-LSP può trasportare solamente pacchetti con un singolo PHB, o di più PHB ma purché condividano lo stesso trattamento da parte dello scheduling ma differiscano in probabilità di perdita; ma non c'è limite al numero di PHB supportati nelle etichette di MPLS, e, per la solita ragione, le informazioni sul PHB devo essere esplicitamente segnalate durante la creazione del LSP. Il L-LSP ha anche il vantaggio della flessibilità del routing, se confrontato con l'E-LSP. Comunque E-LSP è unico e segue un singolo cammino, indipendentemente dalla CoS che esso trasporta. Uno dei principali principi di EuQoS è lasciare la possibilità di calcolare e selezionare AS path differenti a seconda della CoS. Questo è possibile solamente con L-LSP. A causa delle limitazioni di E-LSPe dei requisiti degli EQ-link, cioè che ognuno trasporta traffico per una sola classe di servizio, L-LSP sono preferibili per implementare gli EQ-link.

In ogni caso DiffServ da solo non è sufficiente a soddisfare la QoS desiderata se non è allocato un quantitativo sufficiente di risorse lungo il cammino, indipendentemente dalle tecniche utilizzate. Dal momento che il modello Hard si basa su MPLS, si possono adottare tecniche per ***MPLS-TE – MPLS Traffic Engineering*** specifiche ed efficaci. MPLS-TE permette l'allocazione di risorse, la tolleranza ai guasti e l'ottimizzazione e l'ottimizzazione delle risorse. Tutto questo è realizzato creando LSP lungo i link con risorse disponibili, quindi assicurando cha la banda sia sempre disponibile per un particolare flusso di traffico. Ulteriori ottimizzazioni delle risorse possono essere fatte permettendo ai LSP di non seguire il cammino più corto, se all'interno di quest'ultimo non ci sono risorse a sufficienza. In ogni caso il problema è che MPLS non è cosciente delle classi di servizio, operando sulla banda disponibile come un livello aggregato sopra le classi.

Per superare le limitazioni appena descritte MPLS-TE è stato recentemente standardizzato dal IETF, che ha combinato insieme i vantaggi sia di DiffServ che del TE. MPLS DiffServ-TE diventano quindi MPLS-TE con classi di servizio, permettendo allocazione di risorse con granularità di CoS, e offrendo tolleranza ai guasti di MPLS a livello di CoS. Il risultato è la possibilità di dare garanzie di QoS stringenti e allo stesso tempo ottimizzare l'uso delle risorse. MPLS DiffServ-TE è quindi lo stato dell'arte che combina insieme QoS e TE.

Come precedentemente menzionato gli EQ-link trasportano traffico per una data CoS. Quando un LSP per un dato EQ-link deve essere stabilito , l'Admission Control deve essere eseguito sulla base della banda disponibile (come configurata per MPLS-TE) nel link di uscita. Per permettere una corretta allocazione delle risorse per ogni CoS, è quindi necessario che il nodo sia capace di prendere una decisione che dipende dalla CoS dell'EQ-link. Comunque MPLS-TE non è sufficiente per gestire questo processo, poiché non è a conoscenza dell'esistenza delle classi di servizio, operando sulla banda disponibile a livello aggregato sopra tutte le classi. Questo può essere un problema minore quando il LSP è unito attraverso un LSP di livello superiore, perché in questo caso solamente LSP appartenenti alla stesso CoS possono essere aggregati e, quindi, l'intera capacità assegnata al LSP superiore è disponibile per i LSP interni. In ogni caso MPLS DiffServ-TE può essere di grande aiuto nel caso generale:

- Quando i LSP superiori sono creati durante il processo di allocazione intra-domino con TE.
- Quando le operazioni di aggregazione e stacking non sono usate: Un caso tipo può essere un link inter-AS, dal momento che in questo caso è ragionevole assumere che i LSP originati da tutti i router di bordo in ingresso e terminanti ad un dato router di bordo di uscita, non estendano al di sopra del link inter-dominio uscente.

Infine l'assicurazione che solamente i LSP che servono la stessa CoS possono essere raggruppati insieme è la prima importante caratteristica d implementare nel

modello Hard. Comunque al fine di garantire la QoS nel cammino end-to-end devono essere effettuate ulteriori controlli. Per prima cosa la banda riservata del LSP al livello più alto deve essere maggiore o uguale alla somma delle bande che devono essere riservate ai LSP che vengono raggruppati all'interno di quest'ultimo. Inoltre come la gerarchia può essere creata ad ogni punto del bordo di un AS, il LSP uscente (interno alla rete dell'operatore) al nodo di uscita ha bisogno di essere modellata, in accordo al pSLS negoziato tra gli operatori A e B, e il LSP in ingresso (interno alla rete dell'operatore B) al nodo di ingesso deve essere sottoposto ad un controllo per verificare la conformità alla politica accettata.

3.4.4 Il modello del PCE

Nei router oggi comunemente utilizzati il calcolo del path on-line viene fatto al primo LSR del cammino, ma questo ha delle limitazioni:

- È un'operazione che richiede un intenso uso della CPU.
- In un contesto inter-AS o inter-dominio il LSR non ha la visibilità completa di tutta la topologia, e non può quindi calcolare il path end-to-end.

Per questo motivo la IETF ha creato il concetto di ***PCE – Path Computation Element*** (ed i relativo ***WG – Working Group***). Il motivo che sta dietro alla creazione del PCE è delegare il calcolo del miglior tunnel, richiestogli dai client, per una data rete ad un server dedicato, cioè il PCE stesso. Il progetto originale del PCE prevedeva di prendere in considerazione soltanto i path intra-dominio e non i cammini inter-dominio; quest'ultimi sono l'obiettivo del WG. Infatti dal momento in cui il PCE può comunicare con altri PCE, essi possono cooperare per calcolare path che attraversano più domini.

L'architettura è composta da tre principali funzioni:

- ***PCE – Path Computation Element***.
- ***PCC – Path Computation Client***.

- ***PCEP – PCE Computation Protocol*** che implementa le comunicazioni tra i due.

Il risultato di questa operazione è un ***ERO – Explicit Route Object***, che è utilizzabile per creare un LSP. Questo significa che il PCE non è situato nel piano di management, e non ha il compito di configurare e gestire i LSP. Piuttosto esso lavora come server per calcolo di cammini. Non ci sono requisiti particolari per la localizzazione di PCC e PCE eccetto che essi non possono essere all'inizio del cammino. Entrambi possono essere situati sia all'interno di un router, che al di fuori come entità a se stante. È anche possibile avere più di un PCE in una rete , in questo modo l'architettura non implica necessariamente un'implementazione centralizzata, ma resta aperta la possibilità di un'implementazione distribuita su vari server. Un PCE ha bisogno di accedere ad in ***TED – Traffic Enginnering Database***; infatti allo stesso modo di un LSR, esso è in contatto con il protocollo di routing TE per conoscere automaticamente la topologia della rete e allo stesso tempo le risorse presenti nei vari link annunciati dal protocollo di routing TE. Infine il meccanismo di discovery dei PCE è basato sempre sul protocollo di routing con capacità di TE. Sono state definite delle estensioni di ***OSPF – Open Shortest Path First*** e ***IS-IS – Intermediate System - Intermediate System*** chiamate rispettivamente **OSPF-TE** e **IS-IS-TE** per annunciare le caratteristiche e trovare gli altri PCE. Il PCE può essere sia statefull che stateless, a seconda dell'implementazione[8].

Il principio di funzionamento del PCE è il seguente: Il PCC invia una richiesta per al PCE per calcolare un LSP dal router A fino al router B. Il PCE calcola il miglior path, basandosi sulla sua conoscenza della rete e sui parametri della richiesta. Nel fare questo il PCE, durante la fase di allocazione, si preoccupa di effettuare anche l'Admission Control, infatti dal momento che il PCE può accedere ad un TED, sa esattamente quante sono le risorse disponibili, sulle quali valuta la

[8] IETF raccomanda il PCE stateless.

richiesta fatta dal PCC. Quando il LSP è stato creato il protocollo di routing aggiorna le risorse disponibili e di conseguenza il PCE aggiorna il TED con i nuovi valori. Se il path end-to-end attraversa più di una sola area o di un solo AS la parte finale del path sarà sotto il controllo di un altro PCE; quindi il PCE che controlla la parte iniziale del path deve contattare il PCE remoto per completare il calcolo del path. Nel caso di differenti aree il primo PCE conosce il PCE che gestisce l'area di backbone, quindi può comunicare con lui per costruire il path. Nel caso di differenti AS il primo PCE non può calcolare per intero il LSP, in quanto non sa quale PCE contattare. Una possibilità è utilizzare il cammino Best-Effort, offerto da BGP, l'altra è fornire direttamente al primo PCE l'intero AS-path per la destinazione. Esistono tre alternative per questa possibilità:

- L'AS-path è offerto dal PCC come parametro opzionale
- Il PCE chiede ad un'altra entità l'AS-path
- Il PCE chiede ai suoi vicini gli ERO per la destinazione e poi sceglie il migliore.

3.4.5 Il calcolo dell'EQ-link

Questa funzione può essere separata in due livelli separati:

- Calcolo del path inter-AS, cioè il migliore cammino a livello di AS fornito per una data CoS ed una data coppia di domini.
- Calcolo del path, cioè il cammino nodo per nodo.

Per questa operazione è richiesto un Admission Control per:

- Controllare la disponibilità di risorse prima di creare il path
- Riservare le risorse quando il path è creato

Il primo path, quello con granularità di AS è calcolato tramite l'interazione dei moduli TERO presenti nei domini confinanti. Questo primo passaggio deve prendere in considerazione gli obiettivi di QoS, la disponibilità di risorse, e i vincoli amministrativi (pSLS) che possono limitare la raggiungibilità della destinazione con la CoS selezionata per l'EQ-link. Il risultato del calcolo del path

inter-AS è uno o più path inter-AS, consistenti in una lista di AS da attraversare. Questa lista viene resa disponibile al modulo TERO che avvia la ricerca del secondo path. Il risultato del primo passo è quindi passato al PCE del primo AS che svolge i seguenti compiti:

- Contatta il PCE nell'AS successivo
- Calcola il path intra-dominio quando riceve la risposta dall'altro PCE
- Restituisce l'ERO object al modulo TERO

Di fatto il calcolo del path è svolto a partire dalla destinazione. Quindi è necessario prima contattare i PCE successivi nel cammino, i quali restituiranno la lista di router di bordo che i PCE precedenti dovranno utilizzare, e così via fino al primo AS.

In conclusione calcolare il path inter-AS ed il path intra-AS sono due fasi separate, le quali coinvolgono componenti differenti (nel primo caso i moduli TERO, nel secondo i PCE), e richiedono differenti tipi di interazioni. Attualmente la creazione del path è avviata dal modulo TERO, in relazione all'ERO object che è stato creato dai PCE.

3.4.6 Calcolo dell'AS-path

Come già affermato il calcolo del path inter-AS deve prendere in considerazione gli obiettivi di QoS, la disponibilità di risorse, e i vicoli amministrativi (pSLS). Nel modello Hard il calcolo del path inter-AS viene eseguito seguendo l'approccio "*Loose source routing*". In altre parole il calcolo del path inter-AS è iniziato dal primo AS, il quale negozia un path inter-AS per l'AS di destinazione contattando i suoi vicini.

Questo approccio ha i seguenti pregi:

- Il primo AS può esercitare il controllo sulle proprietà del path inter-AS, piuttosto che sull'intera sequenza di AS attraversati (che giustifica la frase "*source routing*"). Per esempio può evitare di contattare alcuni AS per motivi di performance o di sicurezza, o per politiche di restrizione.

- Può influire sul protocollo EQ-BGP esistente. Quest'ultimo infatti può offrire più path candidati per la stessa destinazione. Un AS apprende il cammino per una destinazione e negozia la possibilità di apprendere le rotte alternative se necessario. Questo porta ad una soluzione scalabile che è compatibile con EQ-BGP, ed inoltre permette politiche di interazione tra coppie arbitrarie di AS. Inoltre gli AS che partecipano alla negoziazione hanno il controllo su quali cammini alternativi, se ci sono, annunciare ad ogni passo. Questo da al AS di transito il controllo sul traffico che entra sulla rete.
- Dal momento che un AS rappresenta un dominio amministrativo indipendente, e le relazioni di business sono facilmente definite a livello di AS, ci si aspetta che i calcoli dei path inter-AS siano semplici e scalabili.
- Il primo LSR non è l'unico che conosce gli obiettivi di QoS end-to-end e la quantità di traffico per la quale questi obiettivi devono essere soddisfati. Da notare che gli EQ-link sono contraddistinti con una CoS, e dei requisiti di QoS devono essere precedentemente specificati devono essere formulati per le varie CoS.

Quando si richiede creazione di un EQ-link devono essere specificate le seguenti

- AS di destinazione.
- CoS per l'EQ-link.
- Obiettivi di QoS (i.e. IPTD, IPDV e IPLR).
- Specifiche del traffico (i.e. il modello token bucket con il quale il traffico che entra nel dominio attraverso l'EQ-link deve essere paragonato).

In linea di principi o altre informazioni possono essere specificate quali:

- Vincoli inclusivi o esclusivi riguardanti il path inter-AS.
- Vincoli sul massimo numero di passi o AS da attraversare.

Attualmente ci concentreremo sul primo gruppo di informazioni, lasciando politiche più sofisticate a sviluppi futuri.

Un requisito funzionale è fornire al modulo TERO i mezzi necessari per selezionare l'AS successivo al fine di far avanzare il path inter-AS. Questo richiede:

- Un adeguato protocollo per la comunicazione tra i vari moduli TERO negli AS confinanti.
- Un algoritmo di decisione per valutare le possibili alternative, in sequenza o in parallelo.

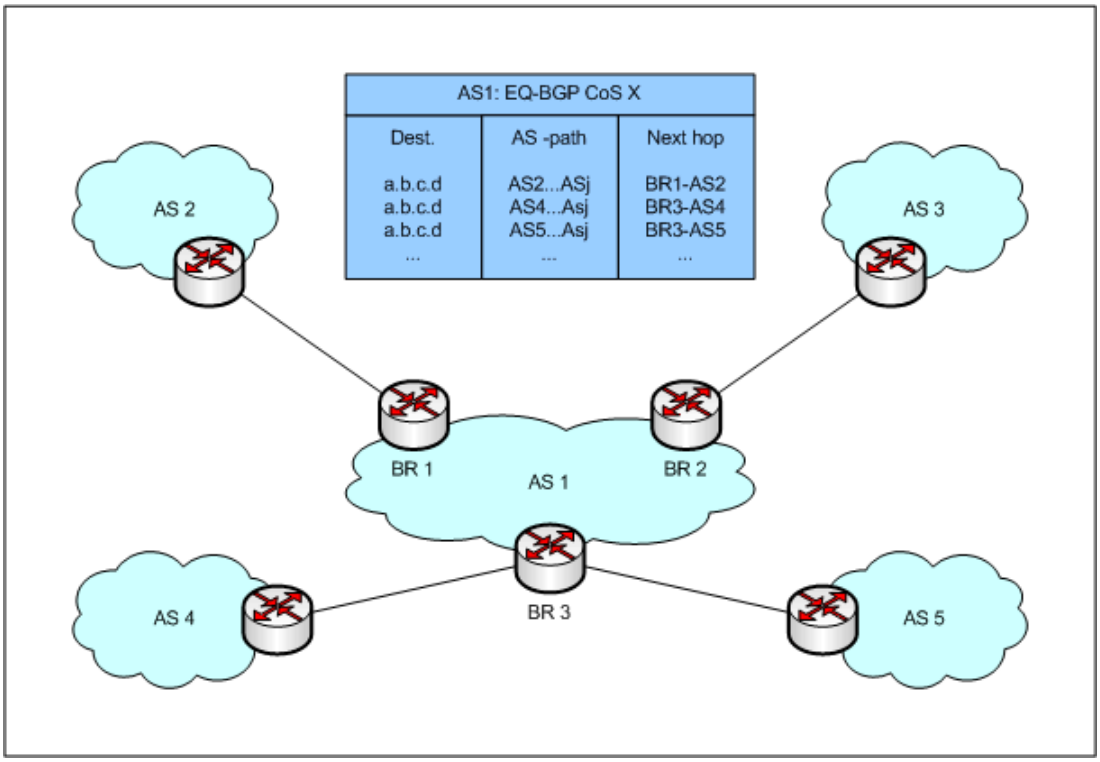

Figura 12 - Calcolo del path inter-AS

Con il rispetto del secondo punto, si può osservare che possiamo investire sullo sviluppo di EQ-BGP presente nel modello Loose. Infatti un AS di solito apprende più di una rotta per una specifica destinazione attraverso EQ-BGP; e l'insieme di rotte è salvato nelle tabelle di EQ-BGP. Di conseguenza la decisione su quali AS confinanti contattare per far proseguire il calcolo del path inter-AS per una data destinazione può essere ristretto a quelli per i quali le update di EQ-BGP per

quella destinazione sono ricevuti. Questo a sua volta implica che gli EQ-link sono creati solamente lungo un path esistente anche nel modello Loose.

Facendo riferimento alla figura precedente [Figura 12], si assume che per esempio AS1 sia il primo AS e che voglia iniziare la creazione di un EQ-link per una CoS X verso un router di bordo T il cui indirizzo IP è **a.b.c.d**. In questo caso il primo modulo TERO scorre la tabella di EQ-BGP (***RIB – Routing Information Base***) per la classe di servizio X per trovare tutte le possibili rotte per il router T. Se il router è raggiungibile, una delle rotte è installata e propagata, mentre le altre non lo sono. Tutte loro descrivono cammini fattibili di tipo Loose per il router T, i.e. forniscono una lista di AS concatenati insieme ai pSLS per la CoS X. Quindi T è attualmente raggiungibile attraverso tutti questi path.

Perciò il primo modulo TERO della catena possiede varie alternative per creare l'EQ-link. A seconda dell'alternativa selezionata, sia il router di partenza, sia il path intra-AS possono essere differenti. Per esempio se l'EQ-link selezionato passa per AS2 allora il router di partenza sarebbe BR1, nel caso invece che l'AS successivo sia AS5 il router di partenza sarebbe BR3.

Il primo modulo TERO apprende dal EQ-BGP una lista di candidati RM negli AS confinanti che esso può contattare per far proseguire il calcolo del path inter-AS. Il primo TERO seleziona uno dei RM che conosce basandosi sulle informazioni di QoS e sulla disponibilità di risorse, e gli invia la richiesta.

Quando un TERO in un AS intermedio è contattato esso svolge esattamente le solite operazioni del precedente. Più precisamente seleziona l'AS successivo tra quelli che annunciano la raggiungibilità dell'ultimo router dell'EQ-link, sempre basandosi su obiettivi di QoS, disponibilità delle risorse all'interno dell'AS. Ovviamente deve essere previsto anche un meccanismo per evitare i cicli (loop) e questo può essere fatto facilmente controllando la lista degli AS in cui già attraversati dall'EQ-link. Da notare che questa tecnica comporta che il primo AS non ha il controllo su che strada seguirà esattamente il path, dal momento che gli AS intermedi contribuiscono al path inter-AS in maniera del tutto autonoma.

Questo è consistente con il modello decisionale decentralizzato cha attualmente è utilizzato su Internet.

Un modulo TERO che è contattato per far proseguire il calcolo del path inter-AS per un EQ-link può ovviamente rifiutare la richiesta se non ha sufficienti risorse per supportare l'ammontare di traffico, o se comunque i contributi alla QoS eccedono la richiesta. Se questo accade esistono diverse alternative possibili:

- Il calcolo dell'EQ-link viene cancellato e possibilmente riavviato dall'inizio.
- Avviene un rollback per uno o più AS, così che il calcolo del path inter-AS non deve essere riavviato dall'inizio, quando possibile, ma continua par altre rotte.

Quest'ultimo approccio richiede che TERO (in tutti gli AS) tenga traccia dello stato delle richieste in cui è coinvolto.

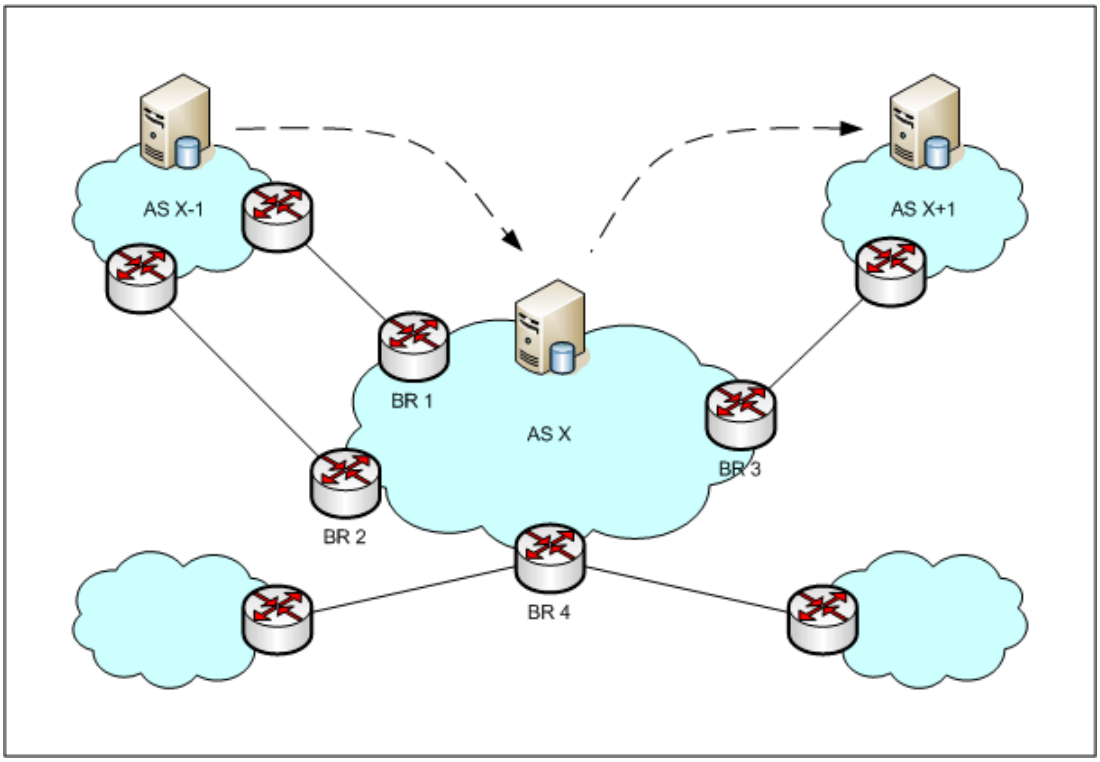

Figura 13 - Calcolo della QoS lungo il path inter-AS

L'overhead che può portare la procedura precedentemente illustrata può essere parzialmente ridotto utilizzando alcuni meccanismi di caching per il calcolo del

path, i quali, come casi estremi, possono implicare che TERO chieda tutti i possibili AS e ne tenga traccia in memoria.

Sottolineiamo che, utilizzando l'approccio precedente, i calcoli collegati ai contributi di QoS di un AS per l'intero ammontare di QoS, sono fatti più precisamente di quelli fatti da E-BGP.

Infatti due AS confinanti di solito hanno uno o più punti di collegamento, dove l'intero numero di punti di collegamento che ha un AS con i suoi AS confinanti , il quale è uguale al numero router di bordo, può essere molto largo, specialmente se l'ultimo è un dominio di transito. Per esempio nella figura precedente [Figura 13] l'AS X ha quattro router, ma possiede due punti di collegamento con l'AS X-1, e uno con l'AS X+1. Nel modello Loose i contributi di QoS dovuti al transito attraverso un dato AS in EQ-BGP erano assunti uguali al caso peggiore tra tutti i path intra-dominio tra due router di bordo. Questo significa che, quando gli update di EQ-BGP sono avvertiti al di fuori dell'AS X gli attributi di QoS sono aggiornati tenendo in considerazione il caso peggiore lungo i path intra-dominio (BR1-BR3, BR2-BR3, BR1-BR4, BR2-BR4, BR3-BR4).

Adesso nel modello Hard, l'AS X viene contattato da TERO presente in AS X-1, per creare un EQ-link verso un certa destinazione. Per decidere se il calcolo del path inter-AS può avanzare verso l'AS X+1 (attraverso il quale, ovviamente è possibile raggiungere la destinazione), Il TERO dell'AS X deve tenere in considerazione i contribuiti di QoS, legati ai path intra-dominio BR1-BR3 e BR2-BR3 a soli. Questo aiuta ad ottenere una migliore utilizzazione delle risorse.

3.4.7 Calcolo nodo per nodo

Recentemente il Working Group dei PCE ha adottato un nuovo draft per estendere il protocollo di comunicazione all'inter-dominio. Con questo nuovo comportamento il PCE diventa adatto per essere utilizzato all'interno del progetto EuQoS. Dal momento che il PCE non ha conoscenza, e non è nel suo interesse, conoscere le rotte inter-dominio, il RM, o più precisamente TERO si deve occupare

di calcolare e offrire al PCE il migliore path inter-AS. Questo guiderà il PCE a calcolare la parte di path intra-dominio del tunnel MPLS e a darla al successivo PCE lungo il path per continuare la creazione del tunnel MPLS. Questa volta il modulo TERO deve solo decidere quali e che ordine di MPLS tunnel è richiesto per creare l'EQ-link per una precisa CoS. Il CAC è eseguito dai PCE quando prende la banda rimanente e capacità di QoS dal protocollo di routing IGP-TE.

3.5 Impatto del modello Hard su EuQoS

3.5.1 Processo di allocazione

Il modello Loose è basato sui seguenti capisaldi:

- Domini confinanti negoziano tra loro pSLS per una data CoS. I pSLS definiscono formalmente:
  - La quantità di traffico che un customer può far passare nel link inter-dominio e le azioni che il provider può prendere su tale traffico se questo non è conforme.
  - I pacchetti di una precisa CoS possono lasciare un AS attraverso un link inter-dominio solamente se esiste un pSLS per quella classe di servizio in quel link inter-dominio. Quindi i pSLS definiscono una topologia virtuale inter-AS, e gli EQ-path sono creati lungo un path iter-dominio che sono connessi in queste topologie virtuali. È particolarmente utile guardare a loro come link virtuali, poiché la connessione tra due AS è generalmente implementata con più di un router di bordo (almeno due) per migliorare l'affidabilità.
- Gli EQ-path sono costruiti durante il processo di allocazione per mezzo di EQ-BGP. Quindi tra una sorgente ed una destinazione è istallato un unico EQ-path per una data CoS, le quali caratteristiche sono note alla sorgente in quanto avvertite tramite gli update di EQ-BGP.

- Il CAC è svolto alla sorgente, quale decide, una volta per tutte, durante il processo di allocazione quando le caratteristiche di QoS del path concordano con la specifica richiesta che deve essere accettata. Infatti gli altri CAC presenti nella rete si limitano a controllare se ci sono sufficienti risorse disponibili, prendendo per vero che le caratteristiche di QoS per quel path sono già state accettate.

Gli EQ-link sono creati come parte del processo di allocazione, e sfruttate durante il processo di invocazione per trasportare il traffico dalla sorgente alla destinazione.

Come già specificato precedentemente, gli EQ-link sono costruiti lungo un EQ-path fattibile dall'AS di partenza all'AS di destinazione. Quando un EQ-link è stato creato i due AS diventano confinanti, anche se solo tramite un link virtuale. Come tali, essi negoziano un pSLS per il quale l'AS sorgente agisce da customer, e l'AS destinazione agisce da provider. Lo schema dei pSLS già utilizzato per il modello Loose non ha bisogno di essere cambiato per essere considerato corretto, anche in caso di AS che sono diventati vicini virtuali attraverso un EQ-link. L'installazione di un pSLS che segue la creazione di un EQ-link fa si che i due estremi dell'EQ-link siano due AS confinanti di EQ-BGP. Questo permette agli update di EQ-BGP di annunciare il link virtuale come fosse uno reale. Per esempio nella figura 1.1 R1.2 e R4.1 stabiliscono un SLS dopo che l'EQ-link è stato creato, R14 annuncia la raggiungibilità della rete B attraverso l'EQ-link.

Quindi gli EQ-path sono creati da EQ-BGP, questa volta, sfruttando l'esistenza di un nuovo link inter-domino virtuale (i.e. EQ-link.), con una QoS conosciuta. Come tali sono costituiti da una sequenza di segmenti del modello Loose ed EQ-link arbitrariamente mescolati.

Il maggiore vantaggio dell'approccio proposto è che il calcolo dell'EQ-path e il CAC rimangono invariati rispetto al modello Loose. Questo significa che possiamo ancora pensare ad un modello di routing inter-dominio decentralizzato, nel quale

gli accordi commerciali (e.g. pSLS) attualmente guidano la creazione di path end-to-end, prendendo priorità sulle considerazioni riguardanti la QoS.

In ogni caso ci sono due potenziali svantaggi. Il primo è che consideriamo il paradigma di BGP come un path tra sorgente e destinazione. Infatti dal momento che un solo EQ-path è disponibile per una data destinazione, implica che è impossibile sfruttare la presenza di più path che viene dalla presenza di EQ-link aggiunti ai link fisici. Più precisamente non dovrebbe essere possibile avere due differenti connessioni dalla stessa sorgente alla stessa destinazione, una lungo l'EQ-path "classico" del modello Loose, e l'altra lungo un path "alternativo" costruito utilizzando gli EQ-link forniti dal modello Hard.

Il secondo svantaggio potenziale del nostro approccio è che il numero di connessioni EQ-BGP che un router deve mantenere attive cresce, portando ad un potenziale problema di scalabilità. Infatti un singolo router (R1.2 e R4.1 nell'esempio in [Figura 10]) può agire come possibile punto di ingresso/uscita per un numero di EQ-link possibilmente grande, il quale richiede un grane numero di connessioni EQ-BGP, portando quindi un eccessivo carico su router. Comunque si può osservare che poiché EQ-link possono essere tra due punti su Internet, e dal momento che un EQ-path può includere più EQ-link in una riga (possibilmente con porzioni di modello Loose nel mezzo), dovrebbe esserci il isogno che un grande numero di EQ-link sia generato da un solo router.

3.5.2 Processo di invocazione

Il Sistema EuQoS offre una soluzione per il CAC end-to-end, che è distribuito lungo i livelli di RM e di RA che include i seguenti passi:

1. Il CAC end-to-end controlla l'esistenza di un path end-to-end. Questo prima è fatto controllando la tabelle di EQ-BGP, ricordando che EQ-BGP offre un path con QoS.
2. Il CAC controlla la disponibilità di un path con QoS all'interno del dominio, i.e. tra il router di ingresso e il router di uscita, e controlla

anche la possibilità di soddisfare i requisiti d QoS nel link inter-dominio. Una volta che il path è stato trovato il CAC-RM si limita ad applicare le politiche dell'operatore alla richiesta e ad inviare la richiesta al RA.

3. Il CAC-RA svolge l'algoritmo di CAC specifico per la particolare tecnologia di rete, ed invia la risposta indietro al RM. Nello stesso momento il RA configura fisicamente gli elementi della rete.
4. Se gli algoritmi di Admission Control a livello del RM e del RA accettano la chiamata, il RM invia la chiamata al RM successivo. Quale sia il RM successivo è dedotto dalle informazioni date quando il primo RM ha selezionato l'EQ-path appropriato.
5. Gli RM successivi svolgono solo i punti 2 e 3 . Il punto 1 viene fatto solamente all'inizio.
6. L'ultimo RM invia un ACK/NACK a quello precedente dopo avere opportunamente configurato le proprie risorse. Per poter terminare il processo.
7. Infine il primo RM invia il risultato all'applicazione che lo ha richiesto.

Come conseguenza il CAC a livello di RM e di RA, controlla, in ogni dominio le potenzialità del path end-to-end offerto da EQ-BGP, rispetto a specificati requisiti di QoS. Alla fine la QoS end-to-end è garantita perché:

- Tutti i vincoli di banda sono stati controllati lungo l'EQ-path
- Tutte le operazioni nei nodi lungo l'EQ-path sono state svolte.

Questo offre all'utente con una connessione end-to-end dedicata appropriata, la possibilità di trasportare il traffico della sua applicazione. Questo algoritmo corrisponde a quello utilizzato nel caso del modello Loose.

3.5.3 EQ-path di un solo EQ-link

Nel modello Hard il caso migliore per la creazione di un EQ-path è il caso in cui quest'ultimo sia composto di un solo EQ-link. In questo caso la connessione tra la rete di accesso sorgente e la rete di acceso destinazione è rappresentata da un solo

link virtuale nel RM. Esso corrisponde ad una connessione punto-punto. Il CAC end-to-end viene quindi eseguito solamente dai RM che controllano le reti di accesso di sergente e destinazione. Veramente, come tutte le rimanenti parti dell'EQ-path sono allocate durante la creazione del MPLS, la banda end-to-end è garantita per una CoS X.

Supponiamo per esempio di voler creare un EQ-path dalla Francia alla Polonia, passando per svizzera e Germania attraverso le rispettive reti nazionali. Si utilizzeranno due livelli di etichette. Adesso, per andare dalla Francia alla Polonia dobbiamo verificare che ci sia sufficiente banda nella rete francese per raggiungere la core network, poi si entra nell'EQ-link, qui l'allocazione e la creazione dei vari LSP garantisce che ci sia una sufficiente banda nella core network. Alla fine è necessario solamente controllare che la banda sia disponibile nella rete di accesso di destinazione (quella polacca). Dunque , il processo di CAC rimane pressoché invariato rispetto al modello Loose; la differenza principale è che c'è solamente un RM sorgente ed un RM destinazione. Tutti i nodi intermedi nella catena sono saltati dal momento che il link virtuale formato dall'EQ-link li aggira.

La complessità del CAC è considerevolmente ridotta poiché:

- Il CAC viene eseguito solamente due volte, senza considerare il numero di AS attraversato.
- Le performance del CAC devono essere dimensionate solamente sui requisiti della rete di accesso, e questo dipende solamente dal numero di utenti connessi alla rete di accesso.
- Lo stato mantenuto nella rete di accesso dipende dalla dimensione della rete di accesso

Comunque le attrezzature di rete nei domini di traffico non devono essere configurate durante il processo di invocazione. L'immagine seguente mostra un semplice scenario con una rete di transito e due reti di accesso.

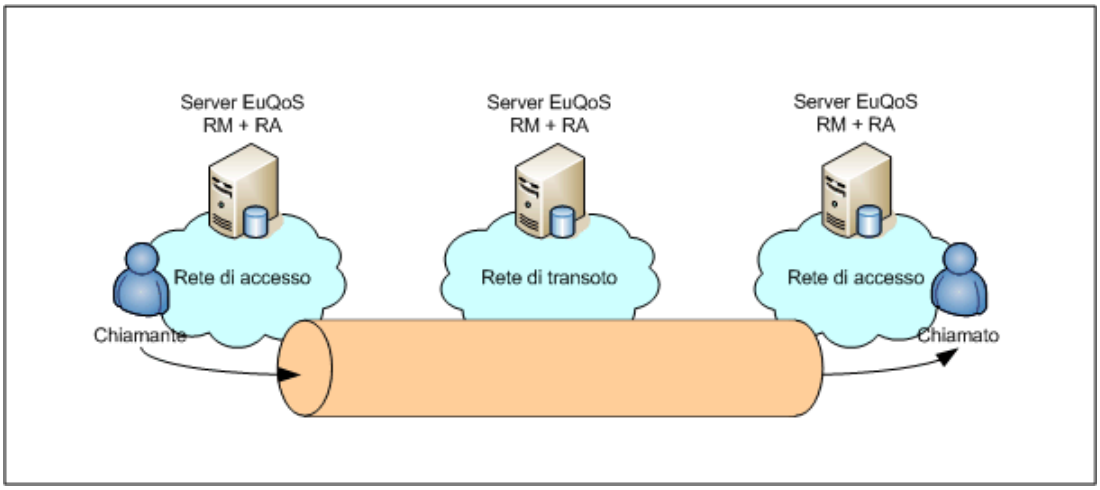

Figura 14 - Scenario del processo di invocazione

Come si può notare dalla figura precedente [Figura 14], un EQ-path con garanzie di QoS deve essere creato tra le due reti di accesso. Nel caso più generale l'EQ-path include zero o più EQ-link, i quali possono non essere contigui. Infatti alcuni AS non implementano MPLS-TE, o peggio non supportano MPLS-TE (e.g. AS con tecnologie WiFi o UMTS). In questi casi il CAC deve essere effettuato durante il processo di invocazione. Gli EQ-link offerti sono infatti una "scorciatoia" tra due AS. Comunque La topologia secondo la quale il RM seleziona il RM successivo lungo l'EQ-path include queste scorciatoie. Quindi mentre nel caso del modello Loose possono essere attraversati dieci AS, richiedendo dieci esecuzioni del CAC, un caso misto con undici AS, con due EQ-link che aggirano quattro AS ciascuno coinvolgerà solamente cinque RM al momento che durante il processo di invocazione i RM degli AS intermedi attraversati da EQ-link non sono chiamati dal CAC. L'algoritmo di CAC rimane pressoché invariato rispetto al caso precedente, anche se viene eseguito un numero inferiore di volte. Quindi questo modello ci permette di sviluppare un sistema EuQoS con una migliore scalabilità.

3.6 L'architettura di riferimento

L'implementazione del modello Hard precedentemente proposto può essere posizionata nella parte più alta dell'architettura di EuQoS relativa al modello

Loose, lasciando sostanzialmente invariato il sistema precedente. L'utilizzo dell'architettura dei PCE di IETF richiede però la modifica di alcuni moduli.

Nella figura seguente [Figura 15] sono mostrati i cambi nel modello architetturale del server di EuQoS.

Confrontata con la precedente versione, il nuovo modello mostra la presenza del server PCE è l'introduzione di un nuovo modulo chiamato RA-MPLS all'interno del RA. All'interno del RM la maggior parte delle funzionalità richieste dal modello Hard sono all'interno del modulo TERO, il quale è responsabile della creazione e gestione degli EQ-link. La figura successiva mostra le modifiche all'interno dell'architettura di TERO in dettaglio.

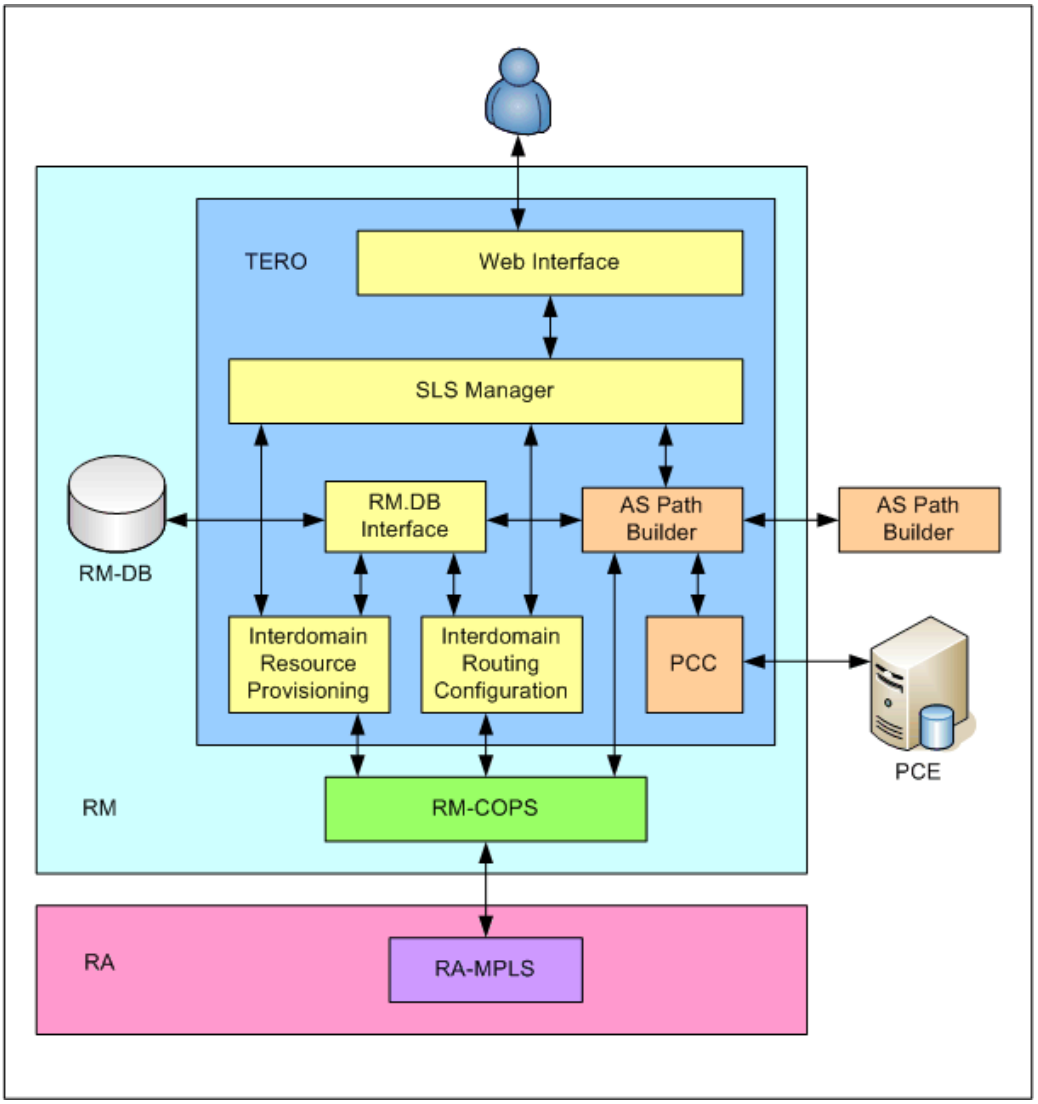

Figura 15 - Architettura software di TERO

La creazione di un nuovo EQ-link può essere richiesta utilizzando l'interfaccia web di TERO per inserire un nuovo pSLS, in maniera simile a come succedeva nel modello Loose. Quando TERO riceve tale richiesta chiama un suo sotto-modulo denominato AS Path Builder, il quale è l'elemento funzionale che ha il compito di determinare la sequenza di AS che devono essere attraversati dall'EQ-link rispettando i vincoli di QoS richiesti. Il calcolo del path è eseguito utilizzando un protocollo distribuito attraverso i vari moduli TERO dei differenti AS, utilizzando un approccio simile a quello del PCEP.

Il path inter-dominio ottenuto è quindi inviato al sotto-modulo PCC, il quale si comporta esattamente come il suo omonimo nell'architettura di riferimento dei PCE [PCEP]. Lungo l'AS-path TERO può anche passare ulteriori parametri (o vincoli) riguardanti la QoS al server PCE. Il PCE processa le richieste ricevute da TERO e restituisce un ***ERO – Explicit Route Object*** il quale identifica il cammino migliore selezionato per stabilire il tunnel MPLS verso la rete di destinazione, seguendo i criteri fornitegli da TERO.

L'oggetto ERO ottenuto è quindi inviato al RA attraverso in modulo RM-COPS. All'interno del RA è definito un nuovo sotto-modulo dipendente dalla tecnologia, chiamato RA-MPLS, il quale ha il compito di configurare e gestire i tunnel MPLS all'interno della rete dell'AS, i.e. invia gli oggetti ERO ai LSR di ingresso.

Il RA-MPLS offrirà un'interfaccia attraverso la quale l'amministratore della rete, ed il modulo TERO potranno creare un tunnel MPLS.

Il RA-MPLS offrirà anche un'interfaccia per creare e cancellare nuove connessioni nei punti di ingresso.

Inoltre il RA-MPLS offrirà un'interfaccia per che verrà utilizzata dal CAC per definire la ***FEC – Forwarding Equivalen Class*** al nodo di ingresso. Questa FEC informa il router di ingresso su come classificare e quindi etichettare il traffico in ingresso per poter inserirlo nel LSP appropriato.

3.6.1 Il protocollo di routing

All'interno del sistema EuQoS si utilizza un protocollo di routing derivato da BGP chiamato EQ-BGP. BGP è un protocollo di routing inter-dominio, dove ogni dominio, nella terminologia di BGP chiamato AS, corrisponde ad una rete IP indipendente sotto un comune amministratore.

L'obiettivo principale di BGP è lo scambio di informazioni riguardanti la raggiungibilità degli AS. Queste informazioni permettono di stabilire cammini tra più AS. Attualmente BGP-4 è utilizzato comunemente si Internet come protocollo di routing inter-dominio.

Le principali regole di BGP sono:

- Stabilire un singolo path tra ogni coppia possibile di AS sorgente – AS destinazione.
- Per ogni AS creare un sink tree.
- All'interno di ogni AS applicare decisioni di routing in base alla politica locale.
- Supportare routing senza classi.
- Permettere l'aggregazione di cammini.

Allo stesso momento le principali limitazioni sono:

- I cammini non sono ottimizzati (nel senso di criteri di performance)
- I cammini sono unidirezionali
- I cammini dovrebbero essere fissi su di un lungo periodo e dovrebbero essere cambiati solo in caso di bisogno
- Supporta solo IPv4.

Da ciò che è stato appena asserito risulta che abbiamo:

- Cammini stabili tra AS.
- Informazioni dettagliate su ogni cammino
- Nessun loop
- Accesso ad informazioni su cammini alternativi se un AS ha più di un link inter-dominio.

I motivi appena accennati mostrano il perché si sia utilizzato EQ-BGP al suo modello classico. EQ-BGP è una versione modificata del protocollo BGP che permette di includere all'interno dei messaggi di update informazioni relative ai servizi disponibili si di una rete, e il livello di QoS offerta lungo il path.

L'approccio di EQ-BGP assume che i messaggi scambiati includano un parametro addizionale, chiamato peso di TE[9]. Questo parametro può essere esprimere banda disponibile, numero di hop, delay massimo ecc. Il valore di questo parametro può quindi modificare le decisioni di routing prese da un determinato dominio. Questo attributo, chiamato ***QOS_NLRI – Quality of Service Network Layer Reachability Information***, permette di scambiare informazioni legate alla QoS quali:

- Packet rate (riservata, disponibile)
- Delay one-way (minimo. massimo, medio)
- variazione del delay tra pacchetti (Jitter)
- Loss rate
- Identificativo del PHB

Queste informazioni sono utilizzate per migliorare le decisioni riguardanti il routing ed in seguito sono inoltrate ai router EQ-BGP confinanti.

3.6.2 Il modulo TERO

La figura [Figura 9] mostra l'architettura software che si trova all'interno del RM. Spostando l'attenzione sul modulo TERO, possiamo osservare che esso interagisce in primo luogo con il Resource Manager DataBase (RM-DB), inoltre effettua scambi di chiamate con i seguenti sotto-moduli di TERO:

- **MMFM**: che ha la funzione di misurare le risorse nella topologie, e rivelare i possibili errori.
- Il **CAC** del dominio: che ha il compito di decidere se una nuova chiamata può o meno essere accettata nel dominio.

[9] Nome originale: TE weight

- Il modulo **Security and AAA**: che ha il compito di gestire l'accesso alle risorse della rete da parte degli utenti (i.e. l'autenticazione), di garantire i servizi e il livello di QoS agli utenti che ne fanno richiesta (i.e. l'autorizzazione), e tenere traccia delle informazioni relative alle connessioni (i.e. l'accaunting).

Leggendo nel RM-DB, TERO può venire a conoscenza di informazioni sulla topologia della rete, e conoscere informazioni statistiche riguardo lo stato delle sessioni. Queste informazioni sono inserite nel database rispettivamente dal modulo MMFM e dal CAC del dominio. Dall'altra parte, se TERO scrive sul database, di fatto invia informazioni al CAC del dominio sullo stato delle risorse nella rete che potranno essere utilizzate per decidere se accettare o meno un nuovo flusso dati proveniente dai link inter-dominio. Dal modulo SAAA TERO ottiene informazioni riguardo gli utenti registrati, che sono utilizzare per calcolare le matrici di traffico a attuali e le previsioni di quelle future. Un'ulteriore comunicazione TERO la tiene con il modulo RM-COPS, il quale rappresenta un'interfaccia che permette a TERO di interagire con il RA per mezzo del protocollo COPS. In particolare RM-COPS è utilizzato da TERO per inviare le informazioni necessarie per configurare le risorse e il protocollo EQ-BGP nei router di bordo.

Se osserviamo TERO possiamo notare che è strutturato in sette moduli, come è sottolineato nella figura seguente:

- **Web Interface**: Offre all'amministratore della rete un'interfaccia web per gestire il modulo TERO.
- **SLS Manager**: Si occupa della gestione di tutte le operazione che devono essere eseguite riguardanti gli SLS.
- **Traffic Analyzer**: Si preoccupa di reperire informazioni sullo stato della rete dalle matrici di traffico e allo stesso temo cerca di prevedere il trend futuro.

- **RM-DB Interface**: permette a TERO di interagire con il gestore del database.
- **Interdomain Resource Provisioning**: Calcola i parametri di QoS e le risorse da allocare per garantire tale QoS (i.e. delay, jitter e loss).
- **AS Path Builder**: Si occupa della creazione di un EQ-link virtuale interagendo con gli altri AS-Path Builder tramite un protocollo distribuito.
- **Path Computation Client**: Offre un interfaccia al modulo TERO per comunicare con il PCE, in modo da ottenere un path inter-dominio.

3.6.2.1 Architettura software di TERO

Il lavoro svolto nell'ambito della tesi si focalizza sulla creazione del sotto-modulo AS-Path Builder, e la progettazione del protocollo di cui esso ne è il principale attore. Nel capitolo successivo ci preoccuperemo quindi di illustrare solamente questo modulo; lasciando i dettagli degli altri moduli alla documentazione del progetto [SPAN], [BISO].

Prima però di descrivere il modulo nel dettaglio è bene fornire alcune informazioni di base sull'architettura software di EuQoS, e quindi del sotto-modulo di TERO chiamato AS-Path Builder.

Il sistema EuQoS è realizzato, quasi nella sua interezza in tecnologia Java, esistono tuttavia alcune sua parti, sul lato client, che per motivi implementativi sono realizzati in tecnologie differenti i.e. C e C++.

La scelta della tecnologia da utilizzare è caduta su Java poiché, sebbene questo linguaggio non offra buoni risultati in termini di prestazioni ed efficienza, offre dei notevoli vantaggi in fase di sviluppo del software tra i quali:

- **Portabilità del software**: Il linguaggio Java[10] è indipendente dalla macchina su cui si utilizza, può quindi essere utilizzato su molteplici ambienti, senza il bisogno di ricorrere a modifiche del codice. Questo

[10] La versione utilizzata è: Java 2 1.4

permette il trasporto di applicazioni da un sistema ad un altro, indipendentemente dal sistema operativo, o dall'Hardware utilizzati. All'interno del progetto EuQoS permette quindi una più semplice e veloce cooperazione tra i partner.

- **Facilità di integrazione**: Attraverso l'utilizzo delle interfacce software, diventa possibile sviluppare i moduli del sistema senza il bisogno di avere a disposizione i moduli necessari. Questo permette una parallelizzazione del lavoro, nonché un fase di integrazione più semplice e veloce.

In un progetto come EuQoS; portato avanti da numerosi partner, sparsi per tutta Europa, è facile intuire come questi aspetti del linguaggio Java siano essenziali per la realizzazione del progetto stesso.

Per quanto riguarda il database utilizzato, la scelta è caduta sulla tecnologia offerta da **MySQL**.

4 Implementazione di TERO

Come accennato nei capitoli precedenti le modifiche software presenti in TERO che sono state effettuate per permettere l'utilizzo del modello Hard si possono prevalentemente osservare nell'aggiunta dei due nuovi sotto-moduli denominati **AS-Path Builder** e **Path Computation Client**. In questo capitolo ci preoccuperemo di fornire i maggiori dettagli possibile sul loro funzionamento, concentrandoci in particolare sul primo dei due, in quanto oggetto del lavoro relativo a questa tesi.

Per la creazione dell'EQ-link sono necessarie due fasi ben distinte tra loro:

- Calcolo del path inter-AS, cioè il calcolo del migliore cammino a livello di AS.
- Calcolo del path intra-AS, cioè il cammino nodo per nodo.

4.1 Struttura software

L'***AS Path Builder*** (***ASPB***), ha il compito di creare un link virtuale inter-dominio, al fine di migliorare l'utilizzo delle risorse necessarie a garantire la QoS, e inoltre rendere migliore la scalabilità del sistema stesso.

4.1.1 Paradigma dell'ASPB

Poiché per calcolare il path inter-AS è necessario un protocollo distribuito l'AS Path Builder deve necessariamente avere i mezzi per comunicare con gli altri ASPB. Questo comporta che la struttura software segua il paradigma client-server. Per essere più precisi al struttura è leggermente più complessa, in quanto, spesso e

volentieri, il server diventa a sua volta un client verso altri. Possiamo quindi immaginare la struttura dell'ASPB come mostrato nella figura successiva.

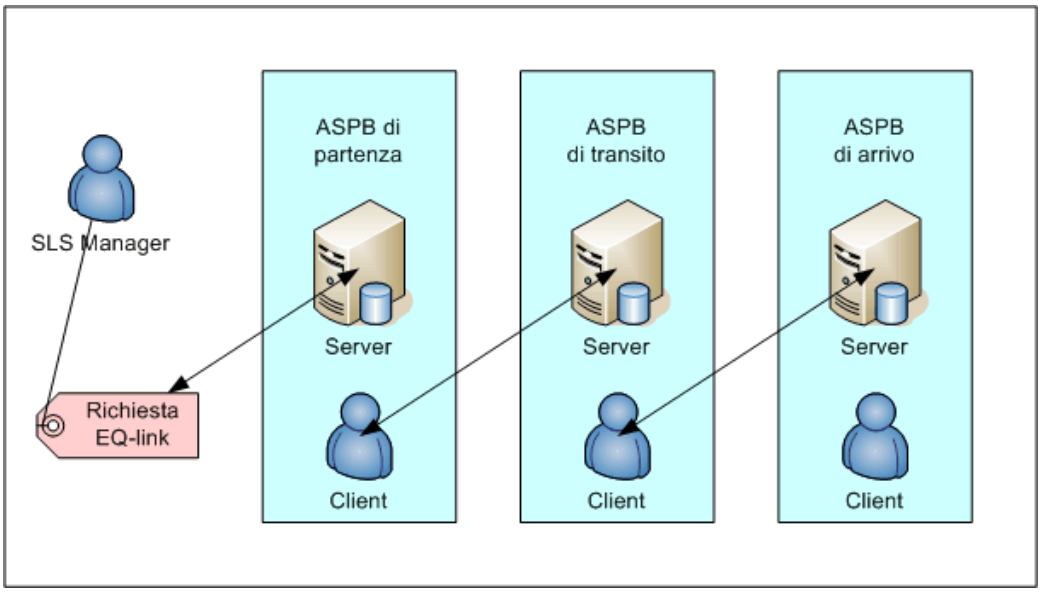

Figura 16 - Paradigma dell'AS Path Builder

Come si può vedere dalla figura, quando il modulo SLS Manager richiede il calcolo di un EQ-link si rivolge direttamente al suo ASPB, più precisamente al server al suo interno. L'ASPB, una volta ricevuta la richiesta, se conosce già la risposta alla richiesta ricevuta (e.g. nel caso in cui non ci siano soluzioni, o siamo già a destinazione, ecc.) risponde in prima persona. In caso negativo l'ASPB utilizza il suo lato client e re-inoltra la richiesta a tutti quegli ASPB che ritiene validi, o utili, per trovare un EQ-path verso la destinazione richiestagli. Un ASPB di transito che riceve una richiesta come la precedente, si comporta nel medesimo modo di quello di partenza, quindi riceve la richiesta, se ne ha la possibilità risponde, e in caso contrario re-inoltra la richiesta. Nel caso in cui ci troviamo ad essere invece l'ultimo degli AS il server può rispondere con affermando che il path è giunto a destinazione.

Una volta raggiunto il termine del path inter-AS, parte una catena di messaggi inversa alla prima, ovvero in direzione dell'AS di partenza per informare quest'ultimo che l'EQ-path è stato trovato.

Naturalmente nulla ci garantisce che tutti i messaggi ricevano una risposta positiva alla loro richiesta. Se per esempio un AS di transito non esistono path verso la destinazione annunciata che sono ritenuti validi, l'ASPB restituisce un messaggio negativo al suo collega a monte nel cammino, e così via fino all'AS di partenza.

L'AS di partenza, una volta ricevuta la risposta (o ricevute se le richieste sono molteplici) gestirà opportunamente il risultato e invierà al modulo SLS Manager un'appropriata risposta.

4.1.2 XML-RPC

Prima di andare ad esplorare nel dettaglio il modulo analizzato è opportuno fare una piccola parentesi sul metodo tramite il quale gli ASPB si scambiano i messaggi.

La tecnica che è stata scelta per lo scambio dei messaggi è ***XML-RPC*** – ***eXtensible Markup Language - Remote Procedure Call***. XML-RPC è una specifica ed un set di implementazione che permettono a software che sono utilizzati su differenti sistemi operativi, con ambenti differenti, di effettuare chiamate a procedure remote su Internet. Queste chiamate a procedure remote sono effettuate utilizzando il protocollo http a livello di trasporto, mentre XML è utilizzato per la codifica. XML-RPC è stato sviluppato per essere il più semplice possibile, e allo stesso tempo permettere di trasportare, processare e ritornare complesse strutture dati.

Un messaggio XML-RPC viene inviato in una richiesta http di tipo post ed il corpo della richiesta è un è un file in formato XML. La procedura richiesta viene eseguita sul server ed in seguito il risultato viene restituito al mittente, sempre formattato in XML. I parametri della procedura possono essere scalari, numeri, stringhe, tipo di dato più complessi e liste di strutture.

Qui di seguito è mostrato un esempio di richiesta XML-RPC:

```
POST /RPC2 HTTP/1.0
User-Agent: Frontier/5.1.2 (WinNT)
Host: cng1.iet.unipi.it
```

```
Content-Type: text/xml
Content-length: 181
<?xml version="1.0"?>
<methodCall>
 <methodName>examples.getStateName</methodName>
 <params>
        <param>
                <value><i4>30</i4></value>
        </param>
 </params>
</methodCall>
```

4.1.3 I processi del modulo

Adesso analizzeremo più nel dettaglio a struttura del modulo, osservandone i processi che lo compongono e le loro interazioni.

L'ASPB è composto da tre processi, di cui due sono da considerarsi principali, ed uno di supporto.

Il primo processo si occupa di ricevere i messaggi del protocollo, inserirli in una coda di messaggi, e tornare nuovamente in attesa di messaggi. In riferimento a quanto detto nel capitolo precedente, possiamo facilmente arguire che questo processo riceve messaggi in formato XML-RPC, ovvero richieste http. Più precisamente questo processo non è altro che un derivato di un server web. Nell'implementazione corrente infatti questo processo è di fatto una servlet attiva all'interno del modulo Jakarta Tomcat di TERO.

Il secondo processo è quello che gestisce i messaggi. Quest'ultimo preleva i messaggi ricevuti dalla coda citata precedentemente, li analizza, ed esegue le opportune operazioni a seconda del tipo ti messaggio che si trova a gestire. Il bisogno di creare questo secondo processo nasce dal fatto che la tecnologia XML-RPC esegue chiamate bloccanti, e tali chiamate possono portare a situazioni di stallo o timeout in alcune topologie, potrebbe infatti accadere che un ASPB invii una richiesta ad un secondo e viceversa, bloccandosi così l'uno in attesa dell'altro. La soluzione dei due thread pone rimedio a questo problema, facendo si che tutti i messaggi vengano ricevuti ed accodati, in modo che il mittente non resti bloccato in attesa dell'esito. I messaggi vengono gestiti sequenzialmente dal secondo

thread, e nei casi in cui si richiede inviare nuovi messaggi, tale operazione è svolta senza la possibilità di incorrere in stallo.

Il terzo processo, ovvero quello definito "opzionale", svolge una funzione, se così la vogliamo chiamare di "recovery". Il processo resta nello stato dormiente per la maggior parte della sua vita; tramite un timer viene svegliato con cadenze regolari, e quando è attivo esegue la pulizia delle tabelle del database relative al protocollo tra ASPB. Più precisamente cancella tutti quei messaggi che sono presenti in memoria non volatile da troppo tempo, lasciando inalterati gli altri. Tutti i messaggi hanno infatti al loro interno un timestamp che identifica il momento in cui sono stati inviati e quindi la durata della loro permanenza nelle tabelle. Di fatto questo ultimo processo non dovrebbe mai creare modifiche alle tabelle, ma esiste sempre la possibilità che un link si spezzi, o una macchina si rompa ecc. per cui il protocollo diventa compromesso. In questi casi nessuno ci assicura che le tabelle vengano opportunamente ripulite se non questo terzo processo. Qui sotto è possibile vedere il dettaglio della struttura software dell'ASPB.

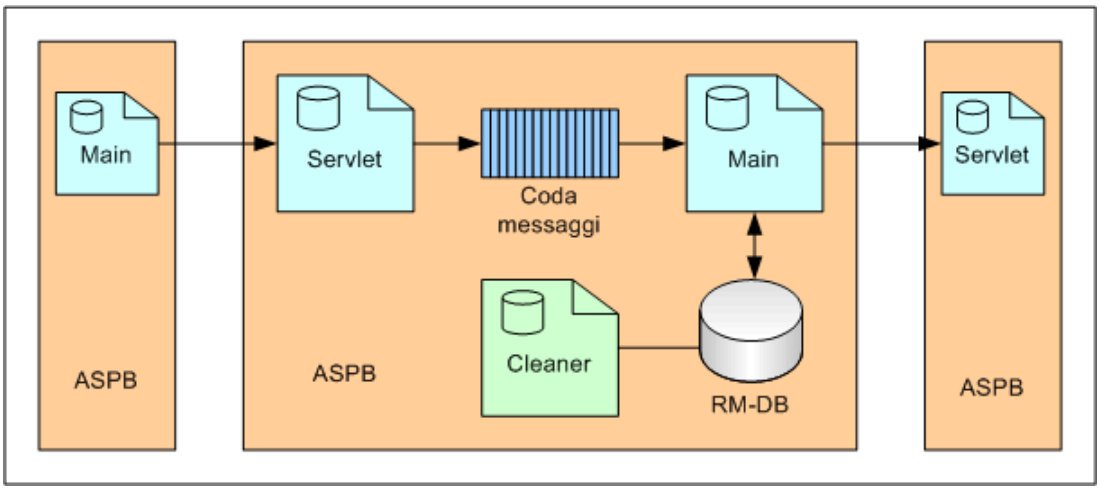

Figura 17 - Thread dell'ASPB

4.1.4 Il protocollo tra ASPB

Questa sezione descrive il protocollo utilizzato dagli AS-Path Builder per la creazione del AS-path inter-AS. Il nome di questo protocollo è AS-Path Builder Protocol.

Le principali caratteristiche di questo protocollo sono:

- Il modello utilizzato è un modello client/server, con l'ASPB che svolge il compito di entrambi a seconda dei casi.
- Il protocollo utilizza TCP come protocollo di livello trasporto, in quanto si trova ad usare il protocollo http per l'invio dei messaggi. Il primo protocollo garantisce l'affidabilità dello scambio di messaggi, il secondo invece offre servizio, o meglio, il mezzo di trasporto utilizzato, per l'invio dei messaggi.
- Per il protocollo non sono fatte assunzioni di nessun tipo riguardo alla sicurezza. Più precisamente i messaggi scambiati non offrono garanzie di autenticità, riservatezza, o simili. Trovandosi il progetto ancora in fase di sviluppo non è infatti richiesto che i messaggi abbiano queste garanzie. Resta comunque il bisogno di tutte le accortezze del caso riguardanti la sicurezza, nel momento in cui il prototipo entrerà in commercio.

4.1.4.1 Messaggi del protocollo

Adesso ci soffermeremo a parlare dei vari messaggi del protocollo e le loro funzionalità.

Ogni messaggio presente nel protocollo è formato dai seguenti campi:

- Header
- Body

L'header è la parte del messaggio che ne descrive le informazioni di argomento più generale. Ha lo stesso formato in tutti i messaggi (naturalmente il contenuto differisce). Il campo body è la rappresentazione del messaggio stesso; al suo interno sono contenuti gli oggetti del protocollo necessari al corretto

funzionamento di quest'ultimo. Il campo body varia a seconda del tipo di messaggio, inoltre varia anche per il solito messaggio a seconda di come si può evolvere il protocollo. Il campo body non è quindi strettamente legato al contenuto dell'header, resta però vero il fatto che il contenuto del body deve essere coerente con quello dell'header al fine di un corretto funzionamento del protocollo.

Tornando alla descrizione dei messaggi, come appena detto il campo header del messaggio è uguale per tutti. È composto da tre campi di tipo che specificano le informazioni principali sul messaggio:

- **Version**: Indica la versione del protocollo. Attualmente esiste solamente la versione 1 ma il campo può essere utilizzato in caso di sviluppi futuri del protocollo.
- **Type**: Rappresenta il tipo di messaggio, e di conseguenza la struttura del corpo del messaggio. Questo campo può assumere cinque valori differenti, i significati dei quali saranno spiegati in seguito:
    1. Request
    2. Response
    3. Error
    4. Cancel
    5. Acknowledge
- **Flags**: Attualmente è un campo opzionale, ed ha sempre valore zero. La sua funzione sarà quella di riscontrare anomalie, o comunque eventi particolari accaduti nel calcolo del path.

A differenza del campo header, il campo body cambia a seconda del tipo di messaggio. Qui di seguito sono illustrati i vari messaggi e la loro composizione. La struttura dei messaggi è ripresa dalla tipologia dei messaggi del PCEP di IETF [PCEP], nel quale per ogni tipo di messaggio è definito un set di regole che specifica l'insieme di oggetti che un messaggio può contenere.

4.1.4.2 Il messaggio Request

Il primo messaggio che incontriamo sia seguendo la lista precedente, che seguendo un ordine logico è il messaggio di Request. Questo messaggio è utilizzato per chiedere ad un ASPB la creazione di un path inter-AS. Il messaggio deve contenere quindi tutte le informazioni necessarie per realizzare il compito. Per prima cosa all'interno del messaggio dovranno essere presenti gli estremi dell'istanza del protocollo e del messaggio stesso, dopo questo è necessario avere informazioni sul cammino che è stato compiuto fino al momento attuale, al fine di evitare loop e cammini troppo lunghi. Altre informazione che sono richieste sono le specifiche del traffico che potrà supportare il link, saranno quindi presenti tutte le specifiche necessarie a caratterizzare il traffico. Infine dovranno essere presenti Come conseguenza di ciò che è appena stato detto il messaggio di Request comprende quattro oggetti:

- ID
- AS Path
- Traffic Specifications
- Guarantees

Quando un ASPB riceve un messaggio di questo tipo il primo controllo che effettua è la verifica dell'arrivo all'AS di destinazione. In caso affermativo si crea il messaggio di risposta: per prima cosa si ricopia il campo ID nel nuovo messaggio, aggiornando opportunamente il PRID. A questo punto si crea un nuovo oggetto di tipo AS Paths con all'interno un unico path, di un solo AS (ovviamente quello corrente). Le specifiche del traffico vengono restituite invariate, mentre tutte le garanzie di QoS sono aggiornate a seconda delle caratteristiche della rete. Una volta creato il nuovo messaggio di Response, quest'ultimo viene inviato al mittente del messaggio di Request corrente.

Assumiamo adesso di non essere all'interno dell'ultimo AS. In questo caso il controllo successivo da effettuare è la presenza o meno di altri messaggi ricevuti per la corrente istanza del protocollo.

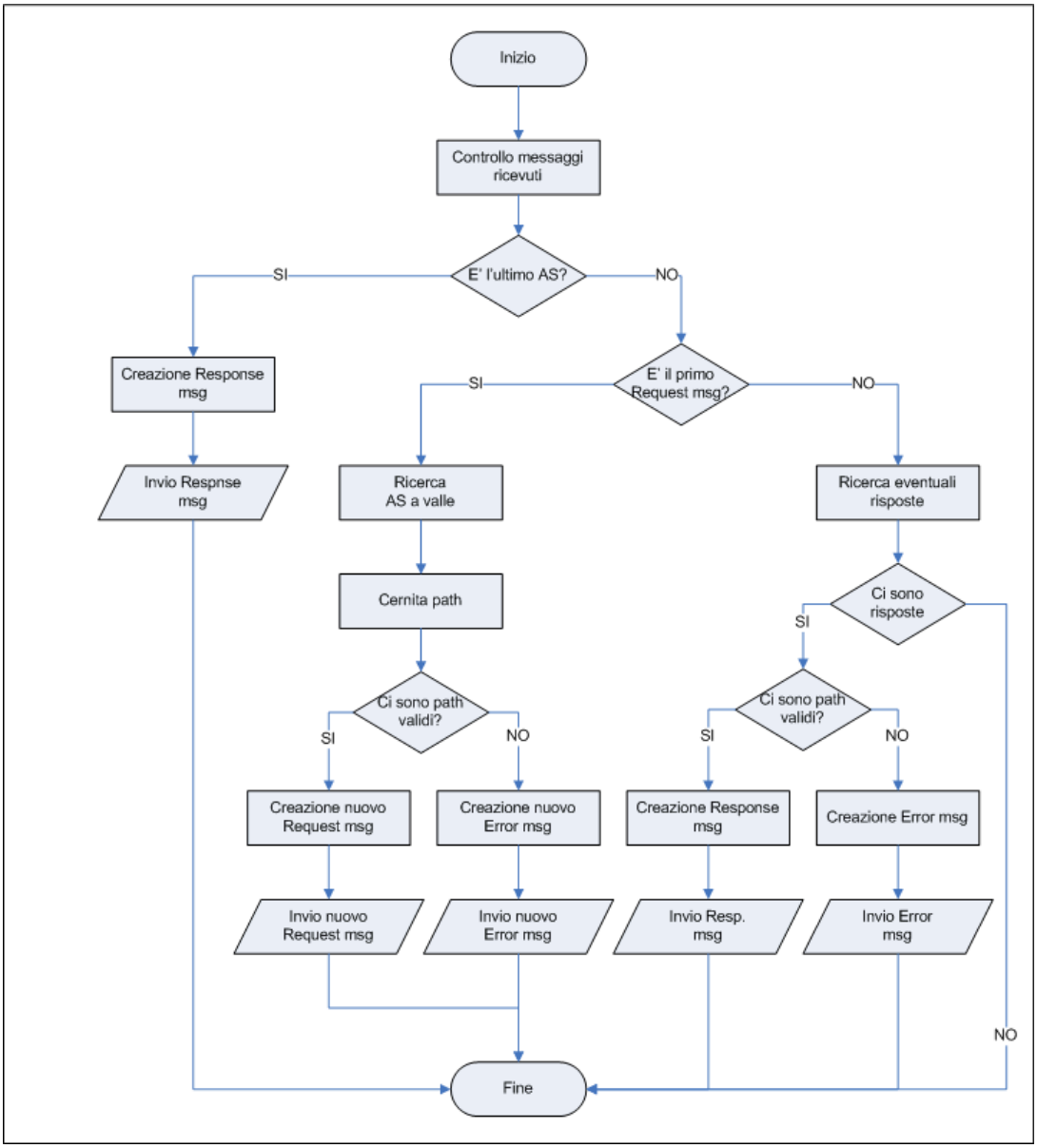

Figura 18 - Gestione del messaggio Request

Se il messaggio ricevuto è il primo con il corrente PCID, sicuramente non si avrà nessun path già pronto a valle che permetta di raggiungere la destinazione prefissata. L'ASPB a questo punto si occupa di ricercare tutti gli AS che sono a valle del suo per i quali esista una rotta verso l'AS di destinazione con la CoS specificata nella richiesta. Tra tutte i cammini trovati si preoccupa poi di escludere quelli che non ritiene validi; questo a seconda di politiche interne al dominio.

A questo punto l'ASPB controlla se esistono rotte valide per la destinazione. In caso affermativo per ognuna di esse crea un nuovo messaggio: più precisamente

copia l'oggetto ID, modificando opportunamente il campo PRID, poi crea un campo AS Path identico a quello ricevuto, ma con l'aggiunta dell'AS corrente alla fine del path temporaneo. Gli altri oggetti vengono copiati direttamente senza subire modifiche. Una vola creato il messaggio viene inviato al relativo AS di destinazione. Se invece non esistono cammini validi ci troviamo di fronte ad una situazione di errore. In questo caso viene creato un messaggio di tipo Error con all'interno il campo ID del messaggio ricevuto (anche qui con le opportune modifiche) e il campo Code ce specifica il codice dell'errore occorso. Il messaggio di Error viene quindi restituito al mittente del messaggio corrente. A questo punto il lavoro dell'ASPB termina.

Se torniamo ad una delle prime scelte, ed ipotizziamo che al momento dell'arrivo del messaggio corrente siano già presenti altre richieste con il solito PCID, l'ASPB controlla se esiste gia la possibilità di ricevere la risposta alla richiesta ricevuta. In caso non ci sia alcuna risposta disponibile, o comunque non ci siano tutte le risposte il lavoro del modulo termina; in caso contrario si esegue il controllo sulla presenza di risposte valide.

Se esiste almeno un path valido si crea un nuovo messaggio di Response: si crea l'oggetti ID sulla base dell'oggetto ID corrente, poi si copia l'oggetto AS paths, aggiungendo ad ogni path al suo interno il numero dell'AS corrente, si copia il campo riguardante le specifiche di traffico ed infine, per quanto riguarda il delle garanzie di QoS, quest'ultime vengono aggiornate in base alle risorse presenti nel dominio, scartando opportunamente tutti i cammini che non rispettano i vincoli forniti nella richiesta. Una volta creato il messaggio esso viene inviato all'ASPB che ha inviato la richiesta.

Nel caso in cui non esista nessun path valido, il messaggio creato è un messaggio di tipo Error, con il campo ID identico a quello che sarebbe stato nel caso della presenza di path validi, ed il campo Code che specifica il tipo di errore che si è presentato.

4.1.4.3 Il messaggio Response

Il messaggio di Response è generalmente utilizzato per rispondere ad un messaggio di Request. Le informazioni che esso trasporta devono essere quindi sufficienti a fornire l'elenco completo del cammini a valle per raggiungere il dominio di destinazione, e le garanzie di QoS che sono state offerte dal punto corrente fino alla destinazione. Sarà quindi presente un oggetto di tipo ID, insieme ad un oggetto che fornisca l'insieme dei path validi trovati. Oltre a questi c'è il bisogno di trasportare informazioni relative alle caratteristiche del traffico trasportato e soprattutto alle garanzie che devono essere offerte al traffico che nel prossimo futuro dovrebbe attraversare il link in costruzione. Seguendo queste informazioni si può affermare che i campi presenti in questo messaggio sono i seguenti:

- ID
- AS Paths
- Traffic Specifications
- Guarantees

Nel momento in cui un ASPB riceve in messaggio di Response per prima cosa, in analogo al comportamento nei confronti del messaggio di tipo Request, si controlla se attualmente ci troviamo nel primo AS del path, ovvero la catena di messaggi di Response ha raggiunto la sorgente.

Se siamo attualmente nel primo AS il controllo successivo da fare è la verifica della presenza di tutti i messaggi di risposta attesi. In caso qualcuno di questi manchi il compito del modulo finisce, nel caso invece che tutti i messaggi di risposta attesi siano arrivati si passa alla fase di interazione con gli altri moduli di TERO; questa sezione è spiegata nei capitoli successivi.

Ipotizziamo invece che non ci troviamo nel primo AS del path. In questo caso si deve controllare se esistono risposte pronte da essere inviate agli AS a monte. Se non c'è alcuna risposta pronta, ovvero non ci siano tutte le risposte provenienti da

valle per poter rispondere a monte, il lavoro del modulo termina; in caso contrario si esegue il controllo sulla presenza di risposte valide.

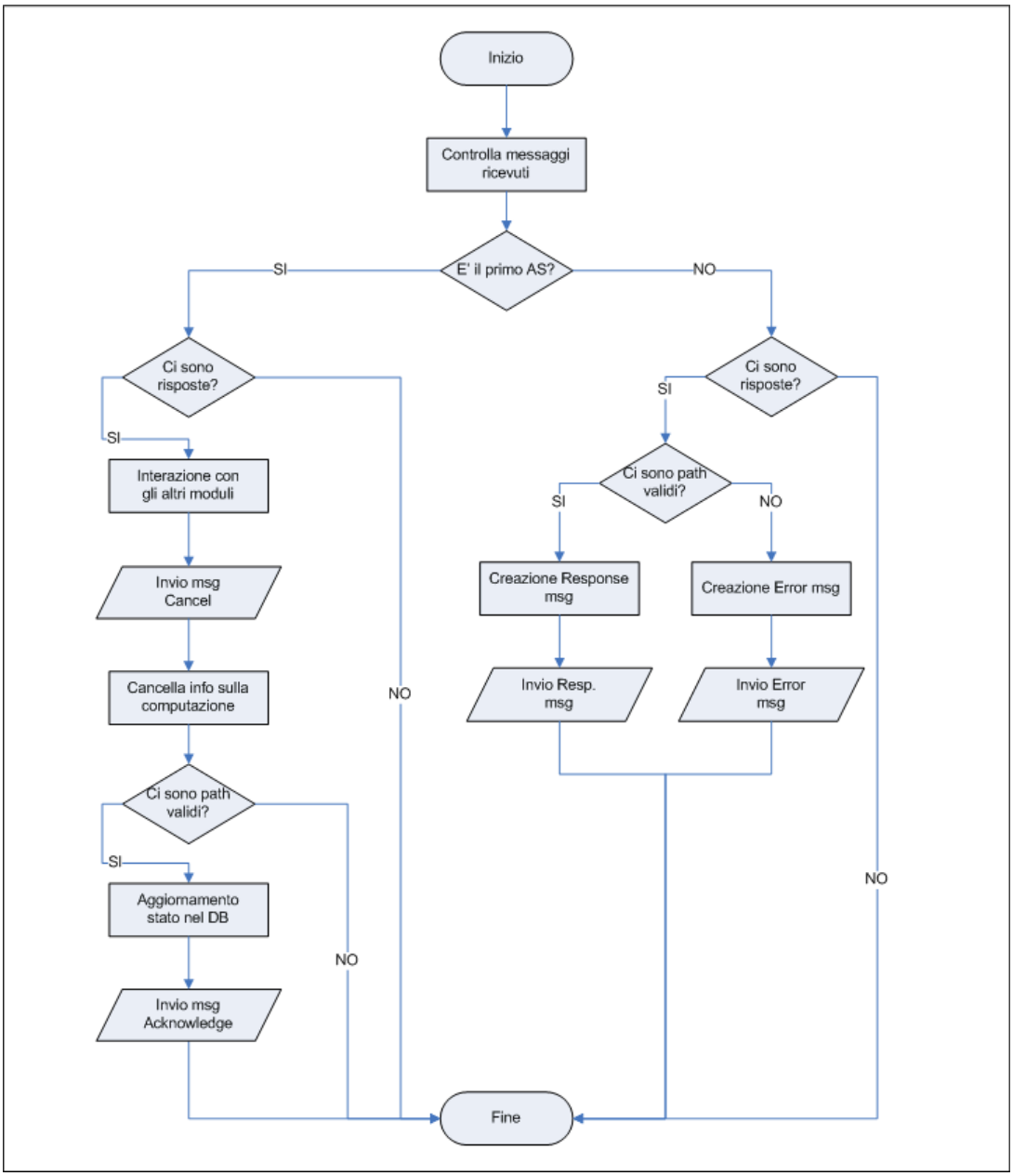

Figura 19 - Gestione del messaggio Response

Nel caso in cui ci sia almeno un path valido si crea un nuovo messaggio di Response: si crea quindi il campo ID sulla base del campo ID del messaggio attuale, impostando opportunamente il PRID. A questo punto si crea una copia del campo AS Paths, aggiungendo ad ogni cammino al suo interno l'informazione riguardante il passaggio per l'AS corrente. Il campo Guarantees viene

opportunamente aggiornato, in base alle risorse presenti nel dominio, e naturalmente scartando i path che, con l'aggiunta del nuovo AS non possono più rispettare i vincoli imposti. Infine il campo riguardante le caratteristiche del traffico viene copiato invariato. Dopo la creazione del nuovo messaggio l'ASPB si preoccupa di inviarlo all'AS che era in attesa di quest'ultimo. Da notare che il messaggio creato viene inviato a tutti gli AS che, con l'aggiunta del messaggio di Response corrente, hanno a disposizione la risposta da parte di tutti i cammini a cui loro erano interessati.

Nel caso in cui non esista nessun path valido, il messaggio creato è un messaggio di tipo Error, con il campo ID identico a quello che sarebbe stato nel caso della presenza di path validi, ed il campo Code che specifica il tipo di errore che si è presentato.

4.1.4.4 Il messaggio Error

Il messaggio di Error è l'analogo del precedente, nel caso però che la risposta inviata non abbia ottenuto buon esito, i.e. nel caso in cui non esistano path, i path presenti non abbiano risorse a sufficienza ecc. Il messaggio deve comprende sia un identificativo dell'istanza del protocollo, sia un codice che permetta di risalire al tipo di errore che si è verificato. Questo comporta che il messaggio contenga i seguenti oggetti:

- ID
- Code

Il messaggio di Error è molto simile al messaggio di Response come concetto; come detto prima i due messaggi servono entrambi a fornire una risposta al messaggio di Request. La differenza principale tra i due è il contenuto che essi hanno. Se questo messaggio si limita a portare un codice di errore, l'altro si preoccupa di portare tutte le informazioni necessarie per la creazione e la selezione dei path diretti verso la destinazione richiesta.

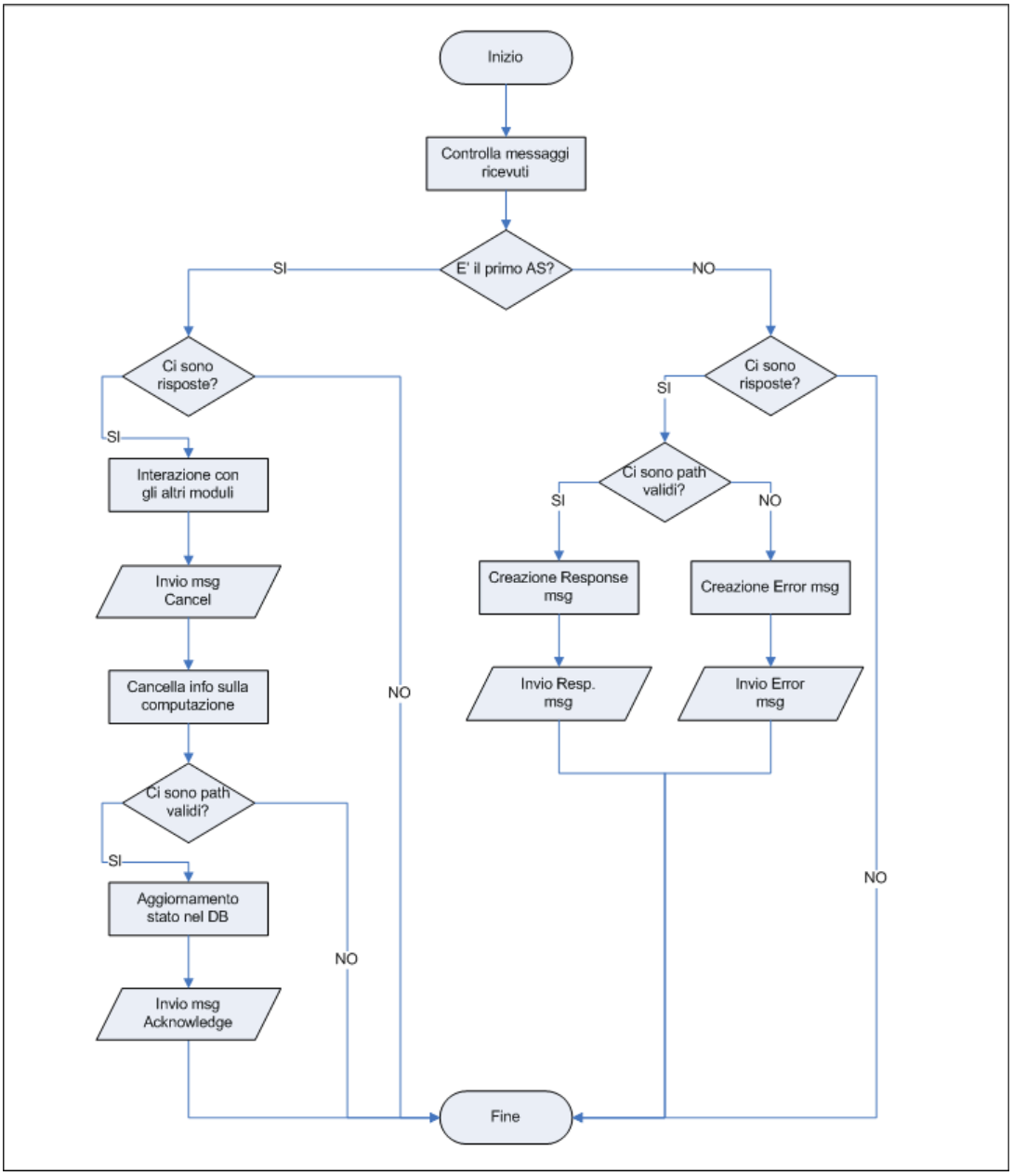

Figura 20 – Gestione del messaggio Error

Anche per il messaggio di Error il primo controllo è sapere se ci troviamo o meno nel primo AS.

In caso affermativo si verifica se tutti i messaggi di risposta attesi sono arrivati. In caso qualcuno di questi manchi il compito del modulo finisce, nel caso invece che tutti i messaggi di risposta attesi siano arrivati si passa alla fase di interazione con il altri moduli di TERO; fase che illustreremo nel dettaglio all'interno dei prossimi capitoli.

Se non ci troviamo all'interno del primo AS si deve controllare se esistono risposte pronte da inviare agli AS a monte Se non c'è alcuna risposta pronta, il lavoro del modulo termina; in caso contrario si esegue il controllo sulla presenza di risposte valide.

Se si hanno a disposizione tutte le risposte utili per determinare il messaggio da inviare a monte la gestione del messaggio di Error diventa identica a quella del messaggio di Error. Per maggiori dettagli si rimanda al capitolo precedente. Esiste tuttavia un particolare da notare: nel caso che si riceva il messaggio di Response si ha la certezza che il messaggio da inviare a monte sarà di tipo Response a sua volta, in quanto almeno il path corrente esiste (ammettendo naturalmente la disponibilità di risorse).

4.1.4.5 Il messaggio Cancel

Il messaggio di Cancel è utilizzato per avvisare tutti gli AS della terminazione dell'istanza del protocollo. Questo significa che gli ASPB possono cancellare tutte le informazioni relative all'istanza in quanto non arriveranno altri messaggi dopo al messaggio di Cancel corrente per questo identificativo. All'interno di questo messaggio l'unica informazione che può servire è l'identificativo quindi al suo interno troveremo un solo oggetto:

- ID

Il messaggio di Cancel è molto semplice da gestire, in quanto richiede solamente due semplici operazioni da parte del modulo. Per prima cosa l'ASPB cerca tutti gli AS che sono stati resi partecipi della corrente istanza del protocollo; dopo di ciò invia a tutti gli AS trovati il medesimo messaggio che è stato ricevuto. Più precisamente il messaggio ricevuto viene copiato nella sua interezza, eccezion fatta per il PRID che naturalmente cambia di messaggio in messaggio. Una volta inviati tutti i messaggi il modulo si preoccupa di cancellare tutte le informazioni presenti nel database che riguardano l'istanza corrente del protocollo. A questo punto il calcolo del path inter-AS può ritenersi concluso.

Si noti che nel caso in cui venga ricevuto un nuovo messaggio di Cancel con lo stesso PCID, il messaggio, come logico che sia, viene semplicemente cancellato dalla coda, in quanto non risultano più informazione nel database relative allo stesso PCID.

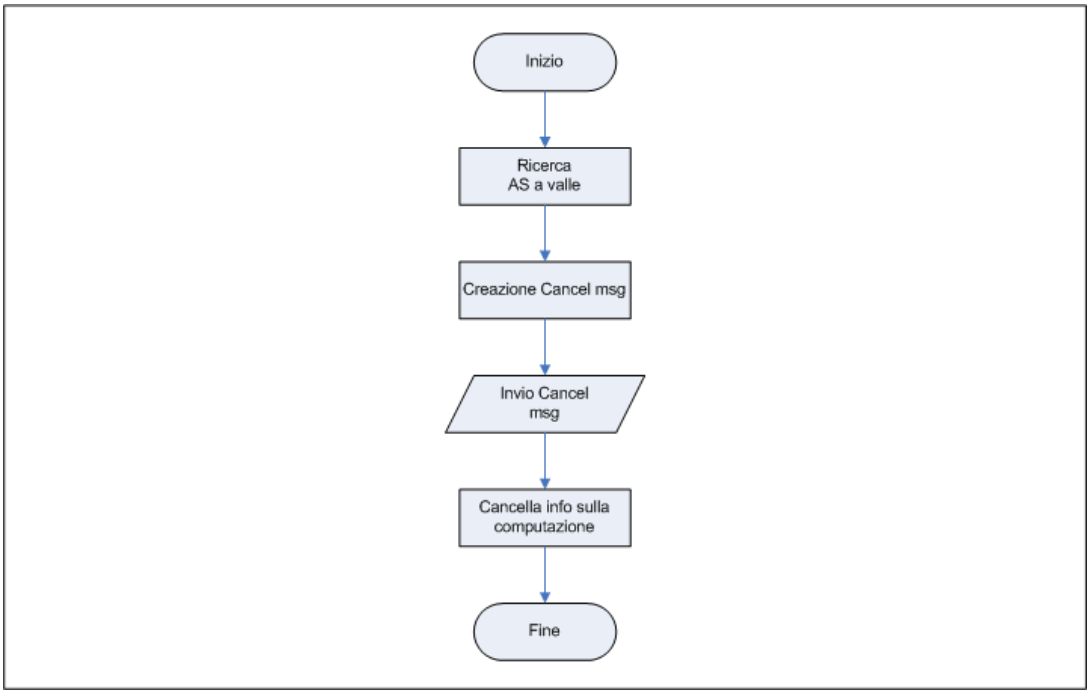

Figura 21 - Gestione del messaggio Cancel

4.1.4.6 Il messaggio Acknowledge

L'ultimo messaggio del protocollo è il messaggio di Acknowledge. Questo messaggio viene inviato ai soli AS che fanno parte del nuovo path inter-AS; si noti quindi che viene inviato solo nel caso di successo, ovvero nel caso in cui il path sia stato effettivamente creato. Questo messaggio permette di informare i vari AS della creazione del nuovo path inter-AS, prima che l'informazione giunga a quest'ultimi tramite l'anello di retroazione formato dal protocollo di EQ-BGP. In questo modo le richieste effettuate immediatamente dopo avranno subito a disposizione una stato della rete coerente con quello reale. Per quanto riguarda la struttura del messaggio, il messaggio è identico al messaggio di Request, con l'unica differenza nella semantica. In questo caso l'AS path non è il path fino a

quel punto, ma l'intero path creato; le informazioni sul traffico non sono quelle previste ma quelle effettive; le garanzie del servizio non sono quelle richieste ma quelle reali. Il messaggio contiene quindi i seguenti oggetti:

- ID
- AS Path
- Traffic Specification
- Guarantees

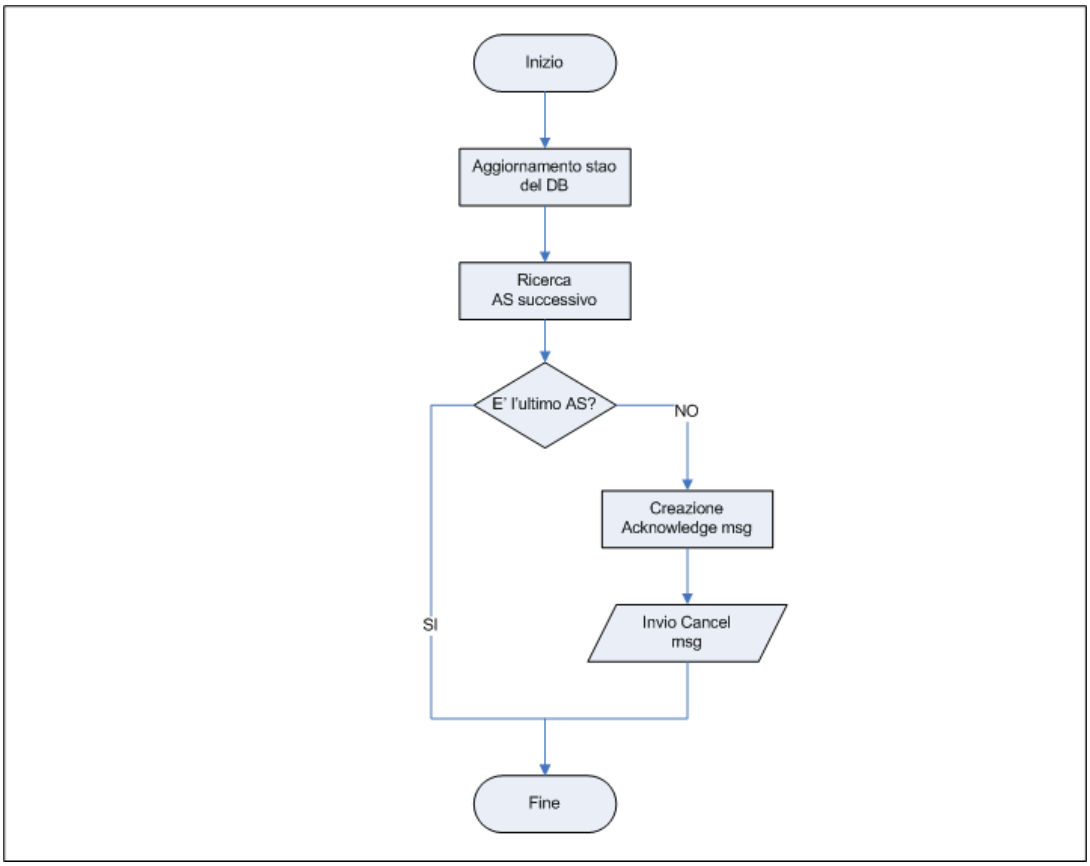

Figura 22 - Gestione del messaggio Acknowledge

Il messaggio di Acknowledge è l'unico messaggio che segue un cammino noto a priori, in quanto segue il path inter-AS appena creato. Quando un ASPB riceve un messaggio di Acknowledge controlla tutte le informazioni relative al nuovo EQ-link stabilito, queste informazioni, opportunamente elaborate, vengono aggiunte al database in modo da poterle riutilizzare nelle istanze del protocollo future. Una

volta gestite le informazioni riguardanti il path creato l'ASPB cerca nel campo AS Path quale sia il prossimo AS nella lista.

Nel caso in cui l'ASPB corrente si trovi nell'ultimo AS del path il lavoro del modulo termina in quanto ci sono altri domini a valle da essere informati. Nel caso contrario invece, in cui non ci troviamo nell'ultimo AS, si crea una copia del messaggio ricevuto, si modifica opportunamente l'identificativo, e si manda questo nuovo messaggio al dominio successivo all'interno del campo AS Path. Qui la gestione del messaggio da parte della rete termina.

4.1.4.7 L'interazione con gli altri moduli

Come accennato nei capitoli precedenti, quando il primo degli AS ha ricevuto tutte le risposte per una particolare istanza del protocollo (questo può capitare solamente nel caso in cui si riceva un messaggio di tipo Response o di tipo Error) inizia ad interagire con le altre parti del modulo TERO. Per prima cosa l'ASPB controlla se esiste almeno un path valido per raggiungere l'AS di destinazione. In caso non ci siamo cammini l'ASPB risponde direttamente al modulo che gli ha inoltrato la richiesta della creazione dell'EQ-link, ovvero al modulo SLS Manager, naturalmente mostrando un risultato negativo. Se invece esiste almeno un cammino valido, il primo dei cammini in ordine di caratteristiche viene fornito ad un PCE tramite l'utilizzo del PCC all'interno di TERO.

La risposta che il PCE fornisce viene nuovamente analizzata dall'ASPB. Se questa è negativa, l'ASPB può inviare una risposta negativa al modulo SLS Manager, oppure nel caso ci siano altri path inter-AS disponibili, provare con il successivo. Se, al contrario, la risposta del PCE è un ERO, il lavoro viene delegato ai LSR presenti nel path, affinché installino l'EQ-link attraverso il protocollo RSVP-TE. Al termine del protocollo, così come nel caso precedente l'ASPB analizza la risposta ricevuta. Se questa è negativa, l'ASPB può decidere nuovamente se inviare una risposta negativa al modulo SLS Manager, oppure nel caso ci siano altri path inter-AS disponibili, provare con il successivo. Se, al

contrario, la risposta è positiva, l'EQ-link è stato creato con successo ed l'ASPB può informare il modulo SLS Manager dell'avvenuta creazione.

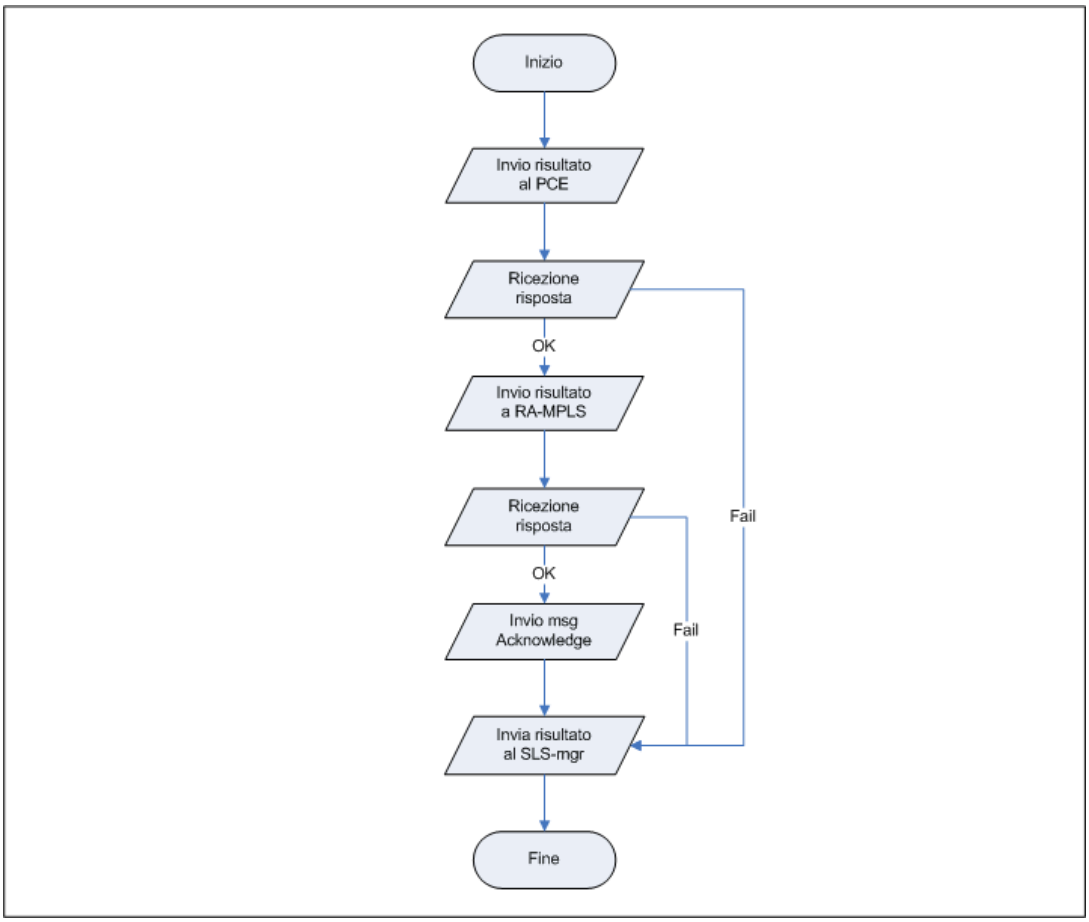

Figura 23 - Interazioni tra l'ASPB e gli altri moduli di TERO

Terminata la fase di interazione con gli altri moduli, l'ASPB si deve preoccupare di informare gli altri domini del termine dell'istanza del protocollo.

Per prima cosa si cercano tutti gli AS confinanti che hanno partecipato all'istanza del protocollo corrente. Si crea quindi un nuovo messaggio di tipo Cancel contenente l'identificativo dell'istanza corrente, opportunamente modificato; ed infine si invia il nuovo messaggio a tutti gli AS selezionati.

Nel caso in cui il protocollo abbia avuto un buon esito si crea inoltre un nuovo messaggio di Acknowledge. Nel nuovo messaggio si inserisce il campo AS Path che contiene il cammino selezionato, il campo Traffic Specification, con al suo interno le informazioni riguardanti il traffico che attraverserà il nuovo link, e il

campo Guarantees, che specifica le garanzie che sono state offerte al link stesso; oltre naturalmente all'identificativo dell'istanza del protocollo.

Il messaggio di Acknowledge viene quindi inviato all'AS successivo all'interno del link appena creato.

4.1.4.8 Oggetti presenti nei messaggi

Gli oggetti qui di seguito elencati sono contenuti all'interno dei messaggi del protocollo di comunicazione. Adesso ne illustreremo le funzioni ed i relativi valori.

Il primo messaggi che troviamo, che è presente in tutti i messaggi, in quanto indispensabile al fine di legare un messaggio alla particolare istanza del protocollo, è l'oggetto ID.

L'oggetto ID contiene tutte le informazioni per identificare univocamente un messaggio; più precisamente contiene:

- Path Computation ID: Identifica la sessione del protocollo. È composta da un intero, ovvero il numero dell'AS che ha fatto la prima richiesta, e da un timestamp che indica il momento temporale a cui è iniziata la sessione.
- Path Reference ID: Identifica un particolare messaggio all'interno di una sessione. Anch'esso come il Path Computation ID è composto da due termini: un interno che rappresenta chi ha creato il messaggio, ed un timestamp che rappresenta quando è stato inviato.
- CoS: La classe del servizio richiesta il nuovo path.
- Start AS: L'AS in cui inizia l'EQ-link virtuale.
- Stop AS: L'AS in cui termina l'EQ-link virtuale.

L'oggetto AS Path contiene la descrizione di un AS path, ovvero l'elenco di tutti gli AS attraversati da un cammino. Questo oggetto può essere utilizzato sia in fase di ricerca del path, per evitare di creare loop all'interno della ricerca stessa; sia a ricerca terminata per informare gli AS del buon esito (si ricorda che il messaggio

di Acknowledge viene inviato solo nel caso in cui il path è creato con successo) del calcolo di un path. La differenza principale tra i due casi è che, nel primo il path è temporaneo e soggetto a cambiamenti, nel secondo caso il path è definito.

L'oggetto AS Paths non è altro che un insieme di oggetti di tipo AS Path. Questo oggetto è utilizzato per restituire i path trovati, tramite questo oggetto infatti è possibile informare della presenza di più path alternativi, e permettere quindi all'AS precedente di scegliere tra più opzioni.

L'oggetto Code rappresenta un codice. Questo oggetto è utilizzato nei messaggi di errore, per specificare il motivo per cui non è stato possibile restituire un messaggio positivo. Il codice si divide un due valori numerici, uno per indicare la tipologia di errore, ed uno per identificare l'errore esatto, all'interno della tipologia.

L'oggetto Traffic Specifications serve a delineare il profilo di traffico che deve passare all'interno di un EQ-link. Questa tipologia è descritta mediante una terna di valori che indica la variabile considerata, (espressa trmite la coppia codice - sottocodice) e un terzo valore che ne indica il valore.

L'oggetto di tipo Guarantees è simile al precedente. Il suo compito è specificare e garanzie che devono essere offerte al traffico che attraversa l'EQ-link che viene creato. Nel caso questo oggetto sia all'interno del messaggio di Acknowledge le garanzie non sono quelle richieste ma quelle fornite. Ogni garanzia è espressa tramite una terna di valori: il primo che identifica la tipologia di garanzia, il secondo che ne specifica il tipo all'interno della tipologia, infine il terzo valore è il valore della garanzia stessa. Per esempio supponiamo che si voglia garantire al traffico un delay medio di 100 ms, un delay massimo di 100 ms ed una loss rate massima dell'1%. Il formato dell'oggetto sarà simile al seguente:

```
<Codice delay>.<Codice delay medio> = 50
<Codice delay>.<Codice delay massimo> = 100
<Codice loss rate>.<Codice loss rate media> = 1
```

5 Conclusioni e sviluppi futuri

Ancora oggi, l'offerta di Qualità del Servizio inter-dominio resta un'operazione alle volte infattibile, alle volte invece, molto complessa e dispendiosa in termini di risorse. Allo stesso momento, la richiesta di garanzie di QoS da parte delle applicazioni multimediali continua a crescere. Si crea quindi sempre di più il bisogno di reti QoS-Aware, che siano in grado non solo di offrire la Qualità del Servizio richiesta loro, ma anche svolgere questo compito nella maniera più efficiente possibile. Per migliorare l'efficienza delle reti con QoS è quindi necessario ridurre al minimo tutto l'overhead generato dal traffico superfluo.

All'interno di questa tesi è stato progettato e sviluppato un sistema che automatizza la creazione di un link virtuale che rende confinanti, i due AS agli estremi del path. Questi link virtuali sono link inter-AS con Qualità del Servizio; la loro creazione implica quindi, un complesso sistema di interazioni tra gli AS coinvolti. È stato prima progettato e poi implementato un protocollo distribuito che, tramite la cooperazione di vari AS, permette la ricerca di un (o più) cammini inter-AS. Gli attori di questo protocollo sono server presenti all'interno della rete, e responsabili delle risorse relative all'AS a cui fanno riferimento. Tali server, per mezzo del protocollo realizzato, cooperano tra di loro per cercare tutti i possibili cammini che legano un punto sorgente a un punto destinazione, filtrando opportunamente quelli che:

- non offrono la classe di servizio specificata
- non permettono di rispettare i vincoli di QoS che sono richiesti
- non sono ritenuti validi per motivi amministrativi

I cammini risultanti vengono poi forniti ad altri server presenti nella rete, il cui compito è affinare il dettaglio del path, passando da un livello di link inter-dominio (e quindi di AS in AS), ad un livello di link intra-dominio (e quindi di router per router).

Una volta ottenuto il path migliore, questo viene installato sulla rete tramite il protocollo RSVP-TE; e al termine di questa operazione il link virtuale è pronto per essere annunciato dai protocolli di routing ed essere attraversato da traffico.

Il lavoro svolto all'interno della tesi fa parte della fase due del progetto EuQoS. Se durante la fase uno si assumeva un modello di tipo Loose, in cui le risorse erano allocate per CoS all'interno di un AS e la creazione di un path implicava la concatenazione di più AS, nella fase due il modello adottato è un modello di tipo Hard, in cui le risorse sono allocate end-to-end, attraverso l'utilizzo di strumenti quali MPLS-TE ed RSVP-TE. Le due modalità non sono tuttavia mutuamente esclusive, anzi, non è pensabile avere la seconda al di fuori dell'ambiente previsto dalla prima. Viceversa la prima soluzione può usufruire dei link creati dalla seconda, per migliorare le proprie prestazioni, senza avere consapevolezza della presenza della Hard.

Per quanto riguarda gli sviluppi futuri riguardanti il modulo ASPB il primo passo da compiere è l'integrazione con l'architettura dei PCE, in particolare con il modulo PCC all'interno di TERO. Altro aspetto da sviluppare in futuro è la creazione di un algoritmo di ordinamento dei path inter-AS che tenga conto delle informazioni di QoS presenti nei vari domini, in modo da avere una lista ordinata con criteri che rispecchiano maggiormente la realtà. Dopo questo un ulteriore sviluppo del progetto è sicuramente la creazione si strumenti automatizzati per il test del sistema in modo da validare quest'ultimo.

6 Riferimenti

[BISO] S. Bisogni "Design and implementation of the Traffic Engineering and Resources Optimization module in end-to-end quality of Service support over heterogeneous netwoks" Luglio 2006.

[EUQS] EuQoS project "*Deliverable D1.1.3: Business models and system design specification*" Agosto 2005.

[EUHM] E. Mingozzi, G. Stea "*Hard model specification*" Gennaio 2007

[MESC] Mescal project "*D1.3: Final specification of protocols and algorithms for inter-domain SLS management and traffic engineering for QoS-based IP service delivery*". Giugno 2005.

[PCEA] IETF "*RFC 4655 – A Path Computation Element (PCE)-Based Architecture*" Agosto 2006.

[PCEP] IETF "*draft-ietf-pce-pcep-03 – Path Computation Element (PCE) communication Protocol (PCEP) – Version 1*" Ottobre 2006.

[PCER] IETF "*RFC 4657 – Path Computation Element (PCE) Communication Protocol Generic Requirements*" Settembre 2006.

[SPAN] A. Spanò "Ingegneria del traffico inter-dominio: progetto e sviluppo di una soluzione per il supporto alla qualita' del servizio in Internet" Dicembre 2005.

[TEQU] Tequila project, "*D1.4: Final architecture, protocol and algorithm specification*" Aprile 2002.